

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería en Sistemas de Información



Trabajo Fin de Grado

Herramientas para la planificación y el seguimiento de proyectos ágiles.

ESCUELA POLITECNICA
SUPERIOR

Autor: Lucía Hurtado de Mendoza Burguillo

Tutor/es: Dra. Carmen Pagés Arévalo

2021



Escuela Politécnica Superior

GRADO EN INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Trabajo Fin de Grado

**HERRAMIENTAS PARA LA PLANIFICACIÓN Y EL
SEGUIMIENTO DE PROYECTOS ÁGILES.**

Autor: Lucía Hurtado de Mendoza Burguillo

Tutor: Dra. Carmen Pagés Arévalo

25 de mayo de 2021

Herramientas para la planificación y el seguimiento de proyectos ágiles

Lucía del Carmen Hurtado de Mendoza Burguillo

25 de mayo de 2021

Agradecimientos

Quiero transmitir mi más sincero agradecimiento a todas las personas que han hecho que mi etapa universitaria sea un recuerdo inolvidable.

Para empezar a las personas más importantes de mi vida: mi madre, mi padre y mi hermano por su dedicación, educación, apoyo y consejos.

Gracias a las amistades que me ha dado la carrera de Ingeniería en Sistemas de Información. Por todos los momentos que hemos pasado juntos, tanto las horas de estudio/prácticas como las de ocio. Es una suerte haberos conocido y poder contar con vosotros para siempre.

Gracias al Erasmus, una de las etapas más felices de mi vida y que siempre quedará en mi recuerdo; por hacerme crecer personal y académicamente además en un año de difícil situación debido a la pandemia. Me ha dado la oportunidad de conocer gente de diversas partes del mundo, poder conocer otras costumbres, comunicarme y estudiar en otro idioma, hacer amigos/as que aunque estén a miles de kilómetros de mí siempre los sentiré cerca.

Por último, agradecer a mi tutora, Carmen Pagés, por guiarme en la última etapa que me queda para completar mi formación universitaria. También a la Universidad de Alcalá por acogerme estos cuatro años. Ha sido una etapa dura, pero muy especial, que recordaré con cariño.

Índice general

| | |
|---|-----------|
| Agradecimientos | iii |
| Resumen | ix |
| Abstract | xi |
| Resumen Extendido | xiii |
| Palabras Clave | xv |
| Glosario | xvii |
| 1 Introducción | 1 |
| 1.1 Marco del proyecto | 1 |
| 1.2 Motivación | 1 |
| 1.3 Objetivo | 2 |
| 1.4 Metodología | 3 |
| 2 Estado del Arte | 5 |
| 2.1 Estudio teórico de las metodologías ágiles | 5 |
| 2.2 Estudio teórico de las metodologías tradicionales | 8 |
| 2.3 Comparación entre las metodologías tradicionales y ágiles | 9 |
| 2.4 Modelos de metodologías ágiles | 10 |
| 2.4.1 Scrum | 10 |
| 2.4.2 Extreme Programming | 12 |
| 2.4.3 Kanban | 16 |
| 2.5 Situación actual de uso de metodologías ágiles en proyectos | 17 |
| 3 Análisis de herramientas para planificación y gestión de metodologías ágiles | 19 |
| 3.1 Jira | 19 |
| 3.2 Trello | 22 |
| 3.3 Wrike | 26 |
| 3.4 Asana | 28 |

| | |
|--|-----------|
| 3.5 Monday.com | 31 |
| 3.6 Tabla comparativa de las herramientas estudiadas | 35 |
| 4 Caso Práctico | 37 |
| 4.1 Definición del proyecto y de las herramientas | 37 |
| 4.2 Plan de ejecución | 39 |
| 4.3 Monday.com | 43 |
| 4.4 Asana | 53 |
| 4.5 Planteamiento final | 64 |
| 5 Conclusiones | 65 |
| 6 Trabajos Futuros | 67 |
| Bibliografía | 67 |

Lista de figuras

| | | |
|------|---|----|
| 2.1 | Ciclo de vida del proyecto | 5 |
| 2.2 | Valores del Manifiesto ágil | 6 |
| 2.3 | Modelo en Cascada | 8 |
| 2.4 | Diferencias entre metodologías tradicionales y ágiles | 9 |
| 2.5 | Ciclo de vida de Scrum 37 | 12 |
| 2.6 | Diagrama de flujo de Extreme Programming | 15 |
| 2.7 | Tablero Kanban sencillo | 16 |
| 2.8 | Utilización de metodologías ágiles en 2019 | 17 |
| 3.1 | Plantillas en Jira | 20 |
| 3.2 | Detalles del Epic | 21 |
| 3.3 | Pantalla principal de Jira Software | 21 |
| 3.4 | Integración con otras herramientas | 22 |
| 3.5 | Pantalla principal de Trello | 24 |
| 3.6 | Detalle tarjetas de Trello | 24 |
| 3.7 | Menú lateral de Trello | 25 |
| 3.8 | Power-ups de Trello | 25 |
| 3.9 | Página principal de Wrike | 27 |
| 3.10 | Colaboración de Wrike | 27 |
| 3.11 | Integraciones de Wrike con otras herramientas | 28 |
| 3.12 | Página principal de Asana | 30 |
| 3.13 | Detalle de la Tarjeta de Asana | 30 |
| 3.14 | Opciones de detalle de la Tarjeta de Asana | 31 |
| 3.15 | Página principal de Monday | 33 |
| 3.16 | Funcionalidades dentro del Espacio de trabajo | 33 |
| 3.17 | Diferentes visualizaciones de la planificación del proyecto | 34 |
| 3.18 | Detalle actividad de Monday | 34 |
| 3.19 | Comparación de las funcionalidades de las herramientas en la versión gratuita | 35 |
| 3.20 | Comparación de las funcionalidades de las herramientas en la versión más completa | 36 |
| 4.1 | Las 8 mejores soluciones software de gestión de proyectos según Finances Online en 2020 | 39 |
| 4.2 | Plan para estudiantes | 43 |

| | |
|---|----|
| 4.3 Creación de nuestro proyecto y sus tableros | 44 |
| 4.4 Plantillas a nuestra disposición | 44 |
| 4.5 Sprint 1 | 45 |
| 4.6 Sprint 2 | 45 |
| 4.7 Sprint 3 | 45 |
| 4.8 Opciones de columnas en el tablero | 46 |
| 4.9 Tablero Kanban | 47 |
| 4.10 Detalle US1 | 47 |
| 4.11 Diagrama de Gantt | 48 |
| 4.12 Diagrama de Gantt exportado a Excel | 48 |
| 4.13 Paneles Monday | 49 |
| 4.14 Ajustes diagrama | 49 |
| 4.15 Gráfico que representa el estado actual del proyecto | 50 |
| 4.16 Tablero de reuniones | 50 |
| 4.17 Tablero con información de los empleados | 51 |
| 4.18 Tablero de reporte de incidencias | 51 |
| 4.19 Integraciones con otras herramientas (A partir del plan estándar) | 52 |
| 4.20 Automatizaciones con otras herramientas (A partir del plan estándar) | 52 |
| 4.21 Plan Business de prueba gratuito | 53 |
| 4.22 Creación del proyecto en Asana | 53 |
| 4.23 Plantillas para la creación de un proyecto | 54 |
| 4.24 Sprint 1 | 55 |
| 4.25 Sprint 2 | 55 |
| 4.26 Sprint 3 | 55 |
| 4.27 Agregar columnas a la tabla | 56 |
| 4.28 Detalle de las tareas de los sprints | 56 |
| 4.29 Marcar tarea de seguimiento | 57 |
| 4.30 Mis tareas | 57 |
| 4.31 Vista tablero | 58 |
| 4.32 Vista cronograma | 58 |
| 4.33 Planificación de Sprints exportado a excel | 59 |
| 4.34 Calendario | 59 |
| 4.35 Paneles | 60 |
| 4.36 Chat de Asana | 60 |
| 4.37 Agenda de reuniones | 61 |
| 4.38 Seguimiento de errores | 61 |
| 4.39 Información de los miembros del equipo | 61 |
| 4.40 Personalizar proyecto | 62 |
| 4.41 Integraciones con herramientas | 63 |
| 4.42 Reglas de Asana | 63 |

Resumen

El presente trabajo fin de grado titulado “Herramientas para la planificación y gestión de proyectos”, realiza un estudio sobre las metodologías que existen para la gestión de proyectos, ágiles y tradicionales, analizando también sus tipos e investigando las diferentes herramientas para llevar a cabo esta planificación y ejemplificarlo con un proyecto real en dos de ellas, mostrando su utilización.

El objetivo final no es solo cumplir con los apartados obligatorios que se requieren en el TFG, sino también saber aplicar estos puntos en un futuro para determinar a qué tipo de proyecto nos enfrentamos y poder planificarlo adecuadamente, con ayuda de la herramienta más eficiente.

Abstract

This thesis entitled “Tools for planning and project management”, deeply analyzes the existing methodologies, agile and traditional, including their types and investigating the different tools in order to do the planning and showing how it works with a real project.

The final goal is not only to accomplish with the mandatory points of the final thesis according to the UAH recommendations, but also determine what kind of project we are facing in the future, planning it correctly using the most efficient tool.

Resumen Extendido

Es un hecho que una buena gestión de proyectos, entendiendo como tal que siguen una metodología para la planificación, seguimiento y dirección de los procesos que conforman el proyecto, es la base para el éxito de este. Además de que no se trata de algo actual, sino que lleva con nosotros mucho tiempo. Sin embargo, las metodologías ágiles surgieron en 2001 con la creación del Manifiesto Ágil que promueven la inclusión del software de una forma más flexibles que las metodologías tradicionales, permitiendo modificaciones durante el desarrollo del proyecto y así intentar obtener el producto final en el menor tiempo posible.

A medida que las metodologías ágiles fueron creciendo, surgieron varios modelos que recogen una serie de pautas para conseguir el objetivo que tienen estas metodologías ágiles. Algunas de las más importantes son eXtreme Programming, Scrum y Kanban. Hoy en día, gracias al crecimiento de la tecnología disponemos de gran variedad de herramientas que nos ayudan a llevar la gestión de los proyectos siguiendo un método ágil. En este trabajo se han analizado 5, entre ellas: Jira, Trello, Wrike, Asana y Monday.com. Saber la parte teórica así como entender los conceptos está genial, pero lo difícil es aplicarlo. Por ello, en este trabajo hemos simulado un proyecto y lo hemos llevado a cabo con la ayuda de dos de estas herramientas.

El objetivo principal que se pretende mostrar a los lectores es que el resultado de que un proyecto salga exitoso depende de muchos factores, algunos los podremos manejar y otros no, pero gracias a las metodologías ágiles cada vez tendremos más capacidad para responder ante situaciones de dificultad. Además, el disponer de la herramienta adecuada nos facilitará todo el proceso.

Palabras Clave

- Gestión de proyectos
- Planificación
- Metodologías
- Software
- Funcionalidad

Glosario

UAH Universidad de Alcalá de Henares.

TFG Trabajo de Fin de Grado.

XP Extreme Programming.

WIP Work In Progress.

US User Story.

CSV Comma Separated Values.

JSON JavaScript Object Notation.

Capítulo 1

Introducción

1.1 Marco del proyecto

Este Trabajo de Fin de Grado (TFG) consiste en la realización del análisis y estudio completo de las metodologías que existen para el seguimiento de los proyectos, y de las herramientas que existen para llevar a cabo esta planificación, desarrollando un proyecto en dos de las herramientas elegidas de las que se han estudiado.

1.2 Motivación

La gestión de proyectos es un conjunto de metodologías para la planificación, seguimiento y dirección de los procesos de un proyecto.

Un proyecto consta de un cúmulo de operaciones destinadas a lograr un objetivo, estableciendo por tanto el inicio y final del proyecto, así como los recursos de los que disponemos. Hoy en día, el fracaso o el éxito de un proyecto depende en su mayor parte de la gestión o la metodología que se lleve a cabo, por eso es muy importante conocer las metodologías que existen y el proyecto que vamos a desarrollar. Los dos tipos de gestión de proyectos a los que vamos a enfrentarnos son la metodología tradicional y metodología ágil.

Para entender la diferencia entre ambas, es necesario hacer una breve descripción de cada una de ellas:

- La metodología tradicional es aquella que marca un plan que hay que cumplir, sin opción a cambios, en la que se antepone la disciplina al desarrollo software, estando todo bien documentado y detallado. Esto hace que los proyectos sean estáticos, ya que, lo que se ha estimado al inicio del proyecto es lo que tiene que cumplirse hasta el final de este. Dentro de las metodologías tradicionales podemos encontrar diferentes tipos que explicaremos a lo largo del proyecto.

- Por otro lado, la metodología ágil destaca por su capacidad de adaptar la forma de trabajo a las condiciones de proyecto consiguiendo por tanto una mayor flexibilidad. A diferencia de la metodología tradicional, los métodos ágiles prefieren software funcionando en vez de documentación extensa. Dentro de la metodología ágil también nos encontramos con varios tipos, pero que detallaremos más adelante.

Lo que buscamos en un proyecto es que esté bien planificado y conseguir los objetivos en el tiempo establecido, para ello necesitamos una herramienta que se haya creado específicamente para esto. Además, existen muchas herramientas para la planificación de proyectos, por eso queremos analizar las que existen, a partir de ahí trabajar con dos de ellas utilizando el mismo proyecto real en ambas y poder sacar conclusiones, viendo sus diferencias y similitudes en cuanto a funcionalidad y usabilidad.

Con esto consideramos que ayudamos a elegir la herramienta que va a ser más adecuada para organizar un proyecto, además de hacer que se invierta el menor tiempo posible en entender cómo funciona la herramienta y poder invertirlo en el propio proyecto. Uno de los motivos que me ha llevado a elegir esta propuesta es en relación con mi puesto de trabajo, ya que tanto el cliente como los integrantes del proyecto buscan el interés común de que el proyecto salga exitoso y que nos haga sentir lo más cómodos posibles con él.

1.3 Objetivo

El objetivo principal del trabajo consiste en el análisis de las diferentes herramientas en la gestión de proyectos ágiles, eligiendo posteriormente las dos más adecuadas para el desarrollo de un proyecto y realizar un cuadro comparativo de ambas indicando detalladamente su usabilidad y funcionalidad a modo de recomendación para futuros trabajos.

El proyecto lo dividimos en sub-objetivos:

1. Estudiar y documentar las metodologías ágiles y tradicionales en la gestión de proyectos.
2. Evaluar los tipos que hay dentro de cada metodología para llevar a cabo un proyecto.
3. Ejemplificar las metodologías ágiles y declarar su situación actual de uso.
4. Investigar las diferentes herramientas de planificación que existen mediante un listado descriptivo extenso.
5. Elegir dos de las herramientas estudiadas, las consideradas más apropiadas para el seguimiento de un proyecto.

6. Definir un proyecto, para llevarlo a cabo en ambas herramientas.
7. Establecer un cuadro comparativo de estas herramientas en cuanto a funcionalidad y usabilidad.
8. Redactar un informe o memoria final.

1.4 Metodología

El método que se aplicará en el trabajo será el más adecuado para el desarrollo del estudio: se llevará a cabo en una revisión bibliográfica profunda sobre la situación actual de las metodologías ágiles, y el análisis de las herramientas que existen para este tipo de planificación, finalmente, se experimentará con dos herramientas diferentes de gestión de proyectos, en las que se desarrollarán en ambas el mismo proyecto para poder realizar una comparación en cuánto usabilidad y funcionalidad.

Capítulo 2

Estado del Arte

2.1 Estudio teórico de las metodologías ágiles

La aparición de las metodologías ágiles ha supuesto un gran cambio en el desarrollo de procesos software que se llevan poniendo en práctica durante 20 años, pero que realmente siempre han existido.

A finales de la década de 1970, [19] [39] gracias a que ya se tenía un conocimiento más amplio sobre la ingeniería del software, se desarrollaron los modelos tradicionales que marcan el inicio y fin de un proyecto pasando por diferentes etapas. Estas etapas lo que consiguen es ajustar los objetivos que se quieren conseguir, para poder satisfacer la necesidad del cliente en el momento al que se ha acordado. A todas estas etapas por las que tiene que pasar un proyecto se le denomina ciclo de vida del proyecto, que pasa por el inicio, planificación, ejecución, seguimiento y control y el cierre del proyecto.



Figura 2.1: Ciclo de vida del proyecto

Las metodologías tradicionales podríamos decir que nacieron prácticamente de la mano con las metodologías ágiles, ya que la idea es la misma pero actualizada y utilizando nuevas tecnologías que se han ido desarrollando a lo largo de la historia. Este concepto se fue introduciendo poco a poco en torno a los años 90. En el 2000 ya se produjo una reunión en la que los participantes de esta expresaban su apoyo a estas metodologías “Light” por Kent Beck, pero nunca se llegó a tomar ninguna decisión concreta, hasta que en febrero de 2001 se convocó otra reunión, esta vez por Robert Cecil Martin, conocido más bien como tío Bob, que era otro de los principales líderes de las metodologías ágiles. A esta reunión acudieron 16 personas más, los pioneros de Extreme Programming (Kent Beck y Ward Cunningham), el propio tío Bob y otros catorce pioneros de la comunidad

ágil. Todos ellos tenían la misma idea, que se refleja en una de las frases que dijo Kent Beck sobre lo que querían conseguir con la creación del Manifiesto ágil “encontrar mejores formas de desarrollar software”. [42][38] El Manifiesto Ágil es un documento donde se recogen 4 valores y 12 principios básicos que siguen todas las metodologías ágiles.



Figura 2.2: Valores del Manifiesto ágil

En ningún momento se considera que los valores de la derecha sean peores que los de la izquierda, sino que se les da mas prioridad en estas metodologías. [13]

- Individuos e interacciones por encima de procesos y herramientas.

Los procesos y las herramientas son de gran ayuda a la hora de mejorar la eficiencia de un proyecto, pero si un proyecto se enfrenta a cambios continuos que requieren de nuevas ideas, espontaneidad o toma de decisiones, necesitamos un equipo que aporte ese conocimiento que impulse el proyecto y la comunicación para satisfacer las necesidades del cliente.

- Software de trabajo por encima de documentación completa.

El entregar una documentación tan extensa acerca del proyecto supone más tiempo que podemos dedicarlo en el software que funcione y que habrá que entregarle al cliente. Por lo que el proceso se agiliza más de esta forma.

- Colaboración con el cliente por encima de negociación contractual.

La constante comunicación con el cliente hace que formen parte del equipo de desarrollo y sea mucho más fácil suplir sus necesidades, además de crear un buen ambiente de trabajo en el que están al tanto en todo momento del estado del proyecto.

- Respuesta al cambio por encima de seguimiento de un plan.

Durante el desarrollo de un proyecto siempre cabe la posibilidad de que surjan cambios a los que el equipo debe estar preparado y adaptarse rápidamente a las circunstancias que se presenten, sin miedo a pensar que eso no era lo que estaba planificado desde un principio.

Como complemento de los cuatro valores que hemos descrito, se redactaron 12 principios en el Manifiesto Ágil: [42]

- i. Satisfacción por parte del cliente al recibir constantes actualizaciones del software.
- ii. Los cambios son bien recibidos en cualquier fase del proyecto.
- iii. Entrega de software que funcione sin mucha demora, desde un par de semanas hasta un par de meses.
- iv. Trabajo en equipo durante el proyecto entre empresarios y desarrolladores.
- v. Brindar el entorno necesario y confianza a los individuos del proyecto que les motive a realizar su trabajo con profesionalidad.
- vi. El intercambio de información cara a cara es la mejor forma de expresión entre las personas.
- vii. La principal medida de progreso es que el software funcione.
- viii. Todos los participantes en el proyecto deberán mantener un ritmo constante, tanto patrocinadores como desarrolladores y usuarios.
- ix. Atender a la excelencia técnica y mantener la calidad del producto para mejorar la agilidad
- x. La simplicidad es esencial para minimizar la cantidad de trabajo a realizar.
- xi. Equipos autoorganizados para cumplir con las mejores arquitecturas, requisitos y diseños.
- xii. Reflexionar sobre como mejorar la efectividad del equipo regularmente.

Las metodologías ágiles por tanto son una filosofía de trabajo que antepone unos valores que suponen una forma diferente de organizarse y trabajar.

El principal objetivo es que el equipo de proyecto sea ágil, para ello entendemos que significa realmente el concepto de agilidad, que según Jim Highsmith este término puede tener muchos significados, pero el más adecuado es [24].

“La agilidad es la capacidad tanto de crear como de responder al cambio para obtener beneficios en un entorno empresarial turbulento.”

2.2 Estudio teórico de las metodologías tradicionales

Las metodologías tradicionales ya mencionadas anteriormente, surgieron antes que las metodologías ágiles. [7] La principal diferencia entre ambas es la utilización de software a la hora de desarrollar un proyecto, por eso también son conocidas como metodologías pesadas, ya que es necesario tener absolutamente todo documentado y planificado desde el inicio del proyecto, con dificultad para realizar modificaciones.

El modelo más antiguo con el que podemos entender la base de la gestión de proyectos es el modelo en cascada (waterfall), ya que se ha considerado el modelo base a partir del cual han perfeccionado dando lugar al resto de metodologías que tenemos hoy en día.

El modelo en cascada consta de 5 fases: requisitos, diseño, implementación, verificación y mantenimiento. Hasta que no queda completada una fase no se puede seguir con la siguiente, esto es lo que a veces supone un retraso en el proyecto cuando hay complicaciones en algunas de las fases. Lo vemos más claro en la siguiente figura :

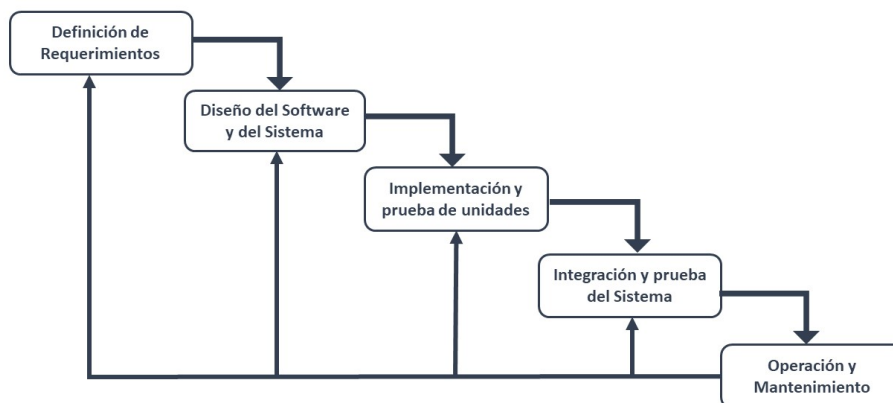


Figura 2.3: Modelo en Cascada

En cada una de estas fases se definen un conjunto de actividades y requerimientos que deben cumplirse antes de pasar a la fase siguiente. Este modelo fue definido por

Winston W. Royce en 1970 como consecuencia de la dificultad del método 'Code and fix' en 1960 [2]. Sin embargo, una de las opiniones que exponía Royce acerca de su propio modelo es:

“Creo en el concepto pero la implementación descrita anteriormente es arriesgada e invita al fracaso.”

Al ser un modelo muy simple se exponía a más fallos, y a veces podía requerir de un rediseño más que de un pequeño arreglo.

A partir del modelo en cascada surgieron otros modelos tradicionales como podemos destacar:

- RUP
- MSF
- Iconix
- WIN-WIN (espiral)

2.3 Comparación entre las metodologías tradicionales y ágiles

La principal diferencia [30] entre las metodologías ágiles y las tradicionales viene dada por la dirección de sus procesos. Una vez ya entrado en contexto en ambas metodologías es importante saber cuáles son sus principales diferencias a modo de cuadro comparativo, así nos facilita tener una visión rápida del tipo de metodología que necesita nuestro proyecto.

| Comparación entre ambas metodologías | |
|--|---|
| Metodologías tradicionales | Metodologías ágiles |
| Los requisitos son definidos al inicio del proyecto. | Reestructuración y mejora de los requisitos cuando sea necesario. |
| El cambio no entra en sus planes. | Filosofía cambiante. |
| Documentación extensa. | Desarrollo software. |
| Prioridad a los procesos. | Prioridad al equipo. |
| El cliente sólo conoce el producto final | El cliente forma parte del proyecto. |
| Equipos grandes. | Equipos reducidos, menos de 10 personas. |
| Coste fijo o poco flexible. | Coste sin concretar al estar expuesto a cambios continuos. |
| Enfocado a proyectos de mayor tamaño. | Centrado en proyectos de menor tamaño. |

Figura 2.4: Diferencias entre metodologías tradicionales y ágiles

2.4 Modelos de metodologías ágiles

Las metodologías ágiles están en el presente, por tanto evolucionarán en el futuro, por eso nos centramos en este TFG en conocerlas más a fondo y trabajar con ellas. Una curiosidad importante que comentar es que a estas metodologías se les puede llamar 'frameworks ágiles' ya que realmente estamos hablando de un conjunto de pautas, consejos o una especie de guía que sigue la empresa o el equipo para que el desarrollo del proyecto sea ágil. Llegados a este punto ya conocemos su historia y filosofía de desarrollo, pero no los modelos que existen. Los modelos son:

- XP - eXtreme Programming
- Scrum
- Kanban
- Lean Management
- Crystal Clear
- DSDM - Dynamic Systems Development Method
- FDD - Feature Driven Development
- ASD - Adaptive Software Development

A continuación exponemos más detalladamente las tres más utilizados:

2.4.1 Scrum

Habréis escuchado hablar mil veces de la metodología Scrum. A día de hoy es la más utilizada por las empresas para el desarrollo de metodologías ágiles. Scrum es un modelo que aporta flexibilidad, agilidad y trabajo en equipo para conseguir mejores resultados. La idea de Scrum surge en 1986 [32] [22] por Hirotaka Takeuchi e Ikujiro Nonaka, pero más tarde en 1995 Jeff Sutherland empleó los principios de esta metodología añadiendo algún matiz por su parte. Finalmente recopilando una serie de reglas y buenas prácticas se creó el Scrum Development Process. Esta metodología pasa por 3 fases [1]:

1. **Pre-game:** en esta fase se detallan cuáles son los objetivos del proyecto, dando prioridad a los que se consideren más importantes para el cliente. Además también se especifican las tareas que se van a realizar, así como las asignación de estas a cada miembro del equipo. El cliente nos proporciona la fecha de entrega del producto final, para ir organizando los Sprints.
2. **Game:** la principal característica de este modelo es la entrega del producto en un periodo de 2-4 semanas para mostrarle al cliente los avances que vamos obteniendo y que el nos de su opinión acerca de ellos. Los sprints son estas entregas para ir evaluando el producto antes de llegar a la fecha final.

3. **Post-game:** por último obtenemos el feedback por parte del cliente acerca de la entrega que se ha hecho en el sprint. Esto es una ventaja ya que el producto se puede mejorar antes de que sea tarde. Nos hace reflexionar acerca de nuestro ritmo de trabajo y qué se puede mejorar.

Algunas de las herramientas [4] que tenemos que conocer para llevar a cabo Scrum son:

- **Product Backlog:** es un documento que contiene los requisitos del producto, las tareas y las asignaciones de estas, cualquier información que sea necesaria para el desarrollo del proyecto.
- **Sprint Backlog:** es una lista donde se detallan los aspectos o tareas que se van a tratar durante el Sprint, así como los objetivos que se han conseguido o que están pendiente de realizarse.
- **Daily Scrum:** se trata de una reunión diaria de entre 10-15 minutos donde cada miembro expone el avance de su tarea.
- **Burn Down:** expone el avance de las tareas y los requisitos que quedan antes de entregar el producto final.

Como en todo equipo hay una organización, y para que esta se cumpla debemos conocer cuáles son los roles y la función que desempeñan dentro de la metodología Scrum, así llegar a los objetivos que se han planeado desde un principio. Los roles más importantes son:

- **Product Owner:** es la persona que se encarga de que el Product backlog quede bien detallado para el desarrollo del proyecto. Solamente debe haber un Product Owner en el equipo, que es el que tomará las decisiones más importantes que el equipo debe seguir para hacer un buen uso de la metodología Scrum. Además, esta persona está en constante contacto con el cliente para conocer sus peticiones y con el equipo para ver los avances.
- **Scrum Master:** es la persona responsable de asegurarse que en todo momento se están siguiendo los principios de la metodología Scrum. Las dos funciones fundamentales son:
 1. Facilitar al equipo de desarrollo la realización de su trabajo mediante la motivación para que se concentren en conseguir los objetivos que deben cumplir.
 2. Eliminar cualquier tipo de obstáculo que pueda suponer un riesgo en el proyecto.
- **Equipo de desarrollo:** es el conjunto de personas que llevan a cabo las tareas que les ha asignado el Product Owner. Deben ser autoorganizados para cumplir con los objetivos del Sprint. Los equipos tendrán un mínimo de 3 personas y un máximo de 9 personas.

- **Stakeholders:** son las personas que ayudan a tomar decisiones del negocio, en cuanto a la financiación o parte económica del proyecto.
- **Cliente:** es el que obtiene el producto final que se ha estado trabajando. Es el que ayuda a definir el Product Backlog. Durante el desarrollo del proyecto está en constante comunicación con el equipo para hacerle llegar su opinión o necesidades.

Scrum es una metodología que se basa en tres pilares fundamentales [27], que son, la **transparencia** a la hora de comunicarse para que todos entendamos lo mismo, la **inspección** de la situación del proyecto y las herramientas de Scrum, por último la **adaptación** a las situaciones las que nos enfrentamos, así como la del producto en el caso de que la idea se aleje mucho de la que el cliente ha solicitado debemos ser capaz de ajustarlo para que lo sea.

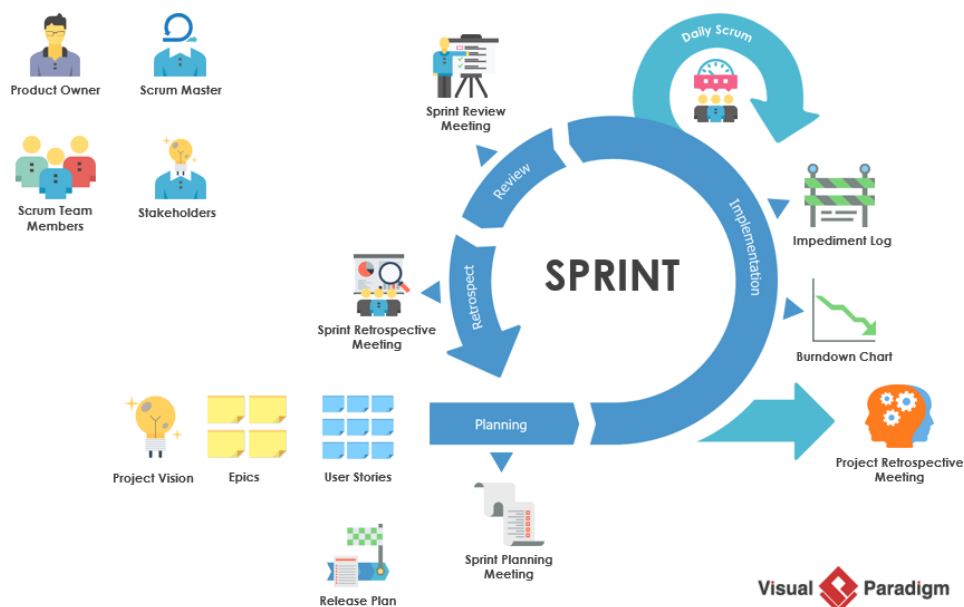


Figura 2.5: Ciclo de vida de Scrum [37]

2.4.2 Extreme Programming

Extreme Programming (XP) es una metodología ágil, enfocada principalmente en hacer pruebas constantemente para reducir el número de errores. Fue desarrollada por Kent Beck en 1996 [18] durante el desarrollo de un proyecto, en el que quiso reunir todas las prácticas ágiles que se habían desarrollada hasta ahora y utilizarlas a un 'nivel extremo'. Tres años más tarde, en 1999 Kent Beck publicó un libro acerca de esta metodología llamado Extreme Programming Explained conocido como "The White Book" [8], en el que se detallan los valores, las variables y los principios que se siguen para conseguir la agilidad en un proyecto, así como los roles que se necesitan en un buen equipo de trabajo. Este modelo está diseñado principalmente para grupos de trabajo pequeños, desde dos

hasta doce personas, en el que se busque la motivación del personal para poder llegar al éxito del proyecto. Por ello, Kent Beck decía: "No soy un gran programador; Solo soy un buen programador con buenos hábitos". No es necesario ser el mejor programando o en la tarea que se te asigne, sino saber llevar a cabo todas estas ideas que se exponen en el libro de Extreme Programming.

Las cuatro variables que conforman el desarrollo de un proyecto son los **objetivos** que se plantean, el **coste** del proyecto, el **tiempo** que va a llevar realizarlo y todos estos dan lugar a la **calidad** del producto. [33]

También define cinco valores muy importantes que son imprescindibles para que haya un buen ambiente de trabajo, potenciando así las relaciones entre los miembros del equipo. Estos valores son:

- La **comunicación** para hacer traspaso de conocimientos durante el desarrollo del proyecto.
- La **simplicidad**, buscando los requisitos realmente esenciales en el proyecto para que la revisión y el mantenimiento sean más sencillos de realizar.
- La **retroalimentación** permite recopilar los comentarios acerca del diseño que se ha definido e implementado para optimizarlo en un futuro.
- El **coraje** significa ser capaz de hacer algún cambio en el proyecto que no sea efectivo y nos haya llevado tiempo hacerlo, este tipo de decisiones requieren valentía.
- El **respeto** entre los miembros del equipo para aceptar las críticas y llegar a alguna conclusión juntos sin crear un entorno incómodo.

Estos cinco valores son los que más destacan sobre unos principios fundamentales, que son los siguientes:

- Enseña a aprender
- Desembozo pequeño inicial
- Jugar a ganar
- Trabajar con los instintos de las personas, no en contra de ellos
- Responsabilidad aceptada

- Experimentos concretos
- Comunicación abierta y honesta
- Adaptación local
- Viaja ligero
- Medición honesta

Obviamente, todos estos principios necesitan llevarse a cabo para obtener un buen resultado del Extreme Programming y que hace que lo diferencie de otros modelos de trabajo. Todo esto se implementa durante unas prácticas, que son 12: [25]

1. **El juego de la planificación (The planning game):** consiste en conseguir una buena planificación mediante la compartición de tareas o responsabilidades entre el cliente y el equipo.
2. **Pequeñas liberaciones (Small releases):** trata de ir haciendo entregas al cliente para saber su feedback.
3. **Metáfora (Metaphor):** es la forma de describir como funciona el sistema.
4. **Diseño simple (Simple design):** como su propio nombre indica, sin complicación, un diseño básico.
5. **Continuas pruebas (Continous testing):** probar continuamente lo que se ha hecho para poder mejorarlo en caso de que se necesite.
6. **Refactorización (Refactoring):** es la reestructuración del código sin que afecte a su comportamiento.
7. **Programación en parejas (Pair programming):** resolución de un problema por dos programadores sobre una misma máquina, así las correcciones son más rápidas y de mayor calidad.
8. **Propiedad colectiva del código (Collective code ownership):** Cualquier miembro del equipo tiene acceso al código en cualquier momento, no es propiedad de nadie.
9. **Integración continua (Continous integration):** cada tarea completada se integra en el sistema y se prueba para resolver los problemas que surjan cuanto antes.
10. **40 horas a la semana(40-hour week):** 40 horas a la semana es el tiempo ideal para dedicarle a trabajar, ya que hacer horas extra conlleva a más errores ya sea por cansancio o aburrimiento.

11. **Historias de usuario (User stories):** el cliente detalla en un documento de la forma más simple la funcionalidad que quieren que el software tenga.
12. **Estándares de codificación (Coding standards):** Dado que el código es accesible por cualquiera, se pretende seguir el mismo estilo de codificación para que sea más fácil de entender para todos.

Además, XP define 7 roles importantes para que un equipo funcione correctamente:

- **El programador:** es el encargado de desarrollar el código del sistema.
- **El cliente:** es el responsable de establecer las funcionalidades que cumplirá el producto y asegurarse mediante la realización de pruebas.
- **El entrenador:** es la persona que motiva al equipo mediante las prácticas que sigue XP y una buena comunicación.
- **El rastreador:** comprueba el tiempo que se ha tardado en completar una tarea o el que se necesita para finalizarla, para así cumplir con la fecha de estimación de entrega del proyecto.
- **El probador:** verifica mediante pruebas que todo está correctamente.
- **El pronosticador:** es el coordinador del proyecto que proporciona los recursos y herramientas necesarias para llevarlo a cabo. También es el que toma las decisiones ante situaciones adversas.

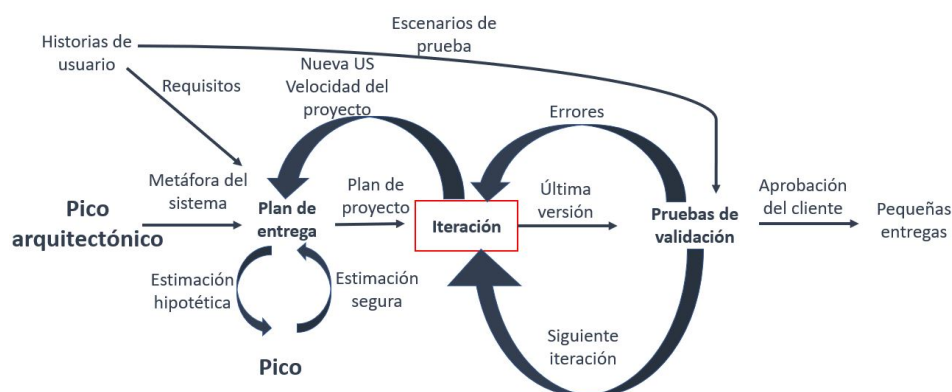


Figura 2.6: Diagrama de flujo de Extreme Programming

El pico arquitectónico es considerado como un Sprint. Se trata de una técnica que consiste en estudiar más a fondo los aspectos que desconocemos de una historia de usuario que nos impiden hacer estimaciones claras y seguras. Con estos picos conseguimos minimizar los riesgos técnicos y aumentar la fiabilidad de la estimación programada.

2.4.3 Kanban

Kanban es otra de las metodologías ágiles que se ha puesto muy de moda. Surgió a principios de los 40 por Taiichi Onho de la empresa Toyota, y como todo en la vida, el nombre Kanban tiene un por qué, surge de la mezcla de dos palabras japonesas que dan el significado de "Tarjetas visuales".

El objetivo de esta metodología es mostrar todas estas tarjetas en un tablero indicando el estado de las tareas y la persona a la que se le ha asignado. Esta metodología se basaba en el Lean manufacturing, que según explicaba Taiicho Onho [26] consistía en *"Producir solo lo necesario, cuando sea necesario y en la cantidad necesaria."* Esto también es denominado "Just in time".

La principal diferencia es que Kanban sigue el denominado Work in Progress (WIP) [23], se declaran las tareas que va haber en el proyecto, pero en el tablero se insertan las más prioritarias y hasta que no se vayan completando las tareas no se añaden las siguientes al tablero. Este método de trabajo hace que nos enfoquemos en las tareas actuales y evitemos cualquier tipo de bloqueo. Gracias a esto creamos un flujo de trabajo continuo y liberamos constantemente tareas al cliente aportando valor al producto.

A continuación mostramos como sería el funcionamiento de este tablero con las tarjetas visuales.



Figura 2.7: Tablero Kanban sencillo

En la columna de Backlog se indican todas las tareas ordenadas por orden de prioridad. Estas tareas se mueven a la columna To do en el momento que se vayan a trabajar, indicando la persona que lo va a realizar. En la columna Doing se añaden las tareas que se están realizando en este momento y se mantienen ahí hasta que se finalice que entonces se moverá a la columna Done. Kanban no tiene unos roles prefijados como en el caso de otras metodologías, contamos con el cliente que solicita un servicio, el equipo que se encarga de llevar a cabo las tareas que se les asignen y un administrador que se encargue de realizar las entregas.

2.5 Situación actual de uso de metodologías ágiles en proyectos

Las metodologías ágiles están en auge dentro de la ingeniería del software, cada vez están más integradas en los equipos de desarrollo de proyectos. Estas metodologías aparte de ayudar a acelerar los procesos de un proyecto permiten el uso de menos documentación y más flexibilidad en los procesos. Hoy en día todo está digitalizado por eso es que las metodologías ágiles y las herramientas que se emplean para llevar a cabo la gestión de los proyectos están creciendo y cada vez tenemos más.

A continuación mostramos un gráfico indicando porcentaje de las metodologías ágiles más utilizadas en el año 2019 [20].

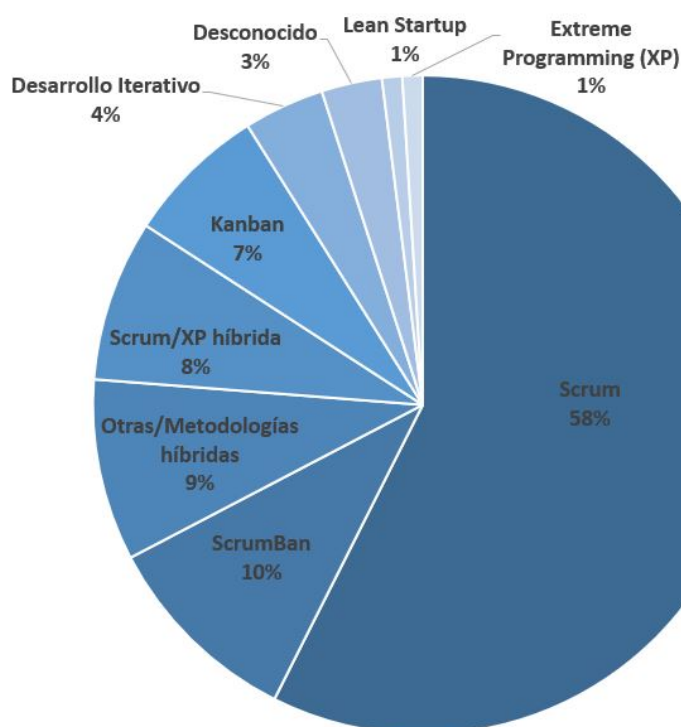


Figura 2.8: Utilización de metodologías ágiles en 2019

Como podemos observar en la figura de arriba el modelo más utilizado entre los equipos de desarrollo es el Scrum. Pero como todo en esta vida, nada es perfecto, por lo que muchas empresas mezclan las ideas de los modelos que más les gustan y lo llevan a cabo. Así se aseguran que de lo que carezca uno de los modelos se lo proporcionará el otro, dando lugar por ejemplo a ScrumBan, mezcla entre Scrum y Kanban.

Uno de los temas que este año nos ha llevado por desgracia a adaptarnos cada vez

más la tecnología es el COVID-19. Por ejemplo, una de las funciones que podría haber afectado a los principios de las metodologías ágiles es que estas ponen especial empeño en que la comunicación entre la plantilla y el cliente sea constante para que se cree esa confianza que hace fluir el proyecto de una manera más fácil, pero gracias a la capacidad de adaptación de agile a cualquier cambio esto se ha podido solventar con reuniones diarias a través de plataformas como Microsoft Teams. A lo que quiero llegar es que, al tener que adaptarnos al trabajo remoto hemos estado familiarizándonos más con las aplicaciones y herramientas que se han utilizado en el desarrollo del proyecto. Además, gracias también a la utilización de las buenas prácticas que expone agile y su continua evolución se puede abordar cualquier desafío.

Capítulo 3

Análisis de herramientas para planificación y gestión de metodologías ágiles

Los tipos de metodologías y sus respectivos modelos ya los hemos estudiado y ahora tendríamos más claro cuál sería el mejor modelo para gestionar el equipo. Nosotros nos vamos a centrar en las metodologías ágiles. Estas metodologías tienen a su disposición infinitas herramientas para ayudarnos a gestionar el proyecto, por eso hemos elegido cinco para profundizar más en ellas. El plan que se ha elegido en todas para analizarlas es el plan gratuito, ya que vamos a tener la primera toma de contacto y con este es suficiente para ver si nos familiarizamos con ellas o no. Finalmente, escogeremos las dos que más nos hayan gustado para gestionar nuestro proyecto.

3.1 Jira

La primera herramienta a analizar es Jira dada su popularidad hoy en día entre los proyectos. Jira es una herramienta que ofrece un conjunto de productos a los equipos para facilitar la organización de sus proyectos. Fue creada en 2002 por una empresa australiana llamada Atlassian. Los productos que ofrece varían en función del servicio que el cliente solicite, estos son: Jira Software, Jira Service Management, Jira Core y Jira Align.

Características principales:

- **Precio:** una ventaja de Jira es que ofrece un período de prueba gratuito durante 7 días y a partir de ahí es necesario elegir uno de los planes según las funcionalidades de las que se quiera disponer. Los planes que ofrecen son: el gratuito que es hasta 10 usuarios, el Standard que es hasta 10.000 usuarios y son 7 dólares por usuario/mes, el Premium que es hasta 10.000 usuarios por 14 dólares usuario/mes y finalmente el Enterprise que hay que consultarlo con el equipo de Jira.

- **Disponibilidad:** es una herramienta fácilmente accesible debido a que está disponible tanto en pc, móvil e ipad para los sistemas operativos IOS, Android, Windows y Linux.
- **Formación:** ofrecen las guías de utilización de Jira en forma de vídeos y seminarios web, así como documentación.

Funcionalidades:

1. **Flujos de trabajo:** es el conjunto de acciones que se realizan a lo largo del proyecto. Por ejemplo, mover una tarea, asignársela a un miembro del equipo... Estos flujos de trabajo son necesarios para ver la progresión del proyecto.
2. **Hojas de ruta:** se trata de un esquema claro y sencillo del producto que está en marcha y así poder ver en todo momento en el estado en el que se encuentra.
3. **Informes:** la creación de informes detallados es una de las funcionalidades que Jira ofrece para mostrar la agilidad con la que se está realizando el proyecto.
4. **Diferentes vistas:** una de las funcionalidades de Jira es facilitar la planificación de un proyecto, es por esto que ofrece tanto la plantilla Kanban o Scrum, ambas son metodologías ágiles.

Interfaz:

La versión que vamos a mostrar a continuación para familiarizarnos con esta herramienta es Jira software gratuita. Al crearnos una cuenta y acceder ya a la aplicación, lo primero que nos pide es crear el proyecto. Antes de crearlo nos pregunta que plantilla queremos utilizar, dándonos a elegir entre la siguientes.



Figura 3.1: Plantillas en Jira

Tras generar el proyecto debemos crear los llamados 'epics', que son las tareas a realizar dentro del proyecto. En cada epic se detalla quién es el que va a realizar la tarea, la descripción de esa misma, el nivel de dificultad y el estado en el que se encuentra. Además, se pueden añadir documentos, vincular tareas y crear incidencias de esa tarea.

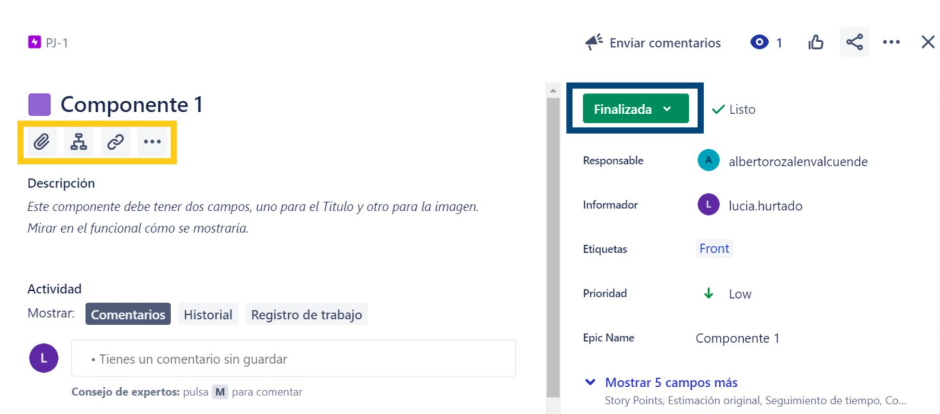


Figura 3.2: Detalles del Epic

Una vez creadas todas estas tareas, vemos la pantalla principal de la aplicación. A la izquierda se encuentra el menú principal con todas las funcionalidades que podemos utilizar, y en la parte superior el nombre del proyecto en el que nos encontramos. Por último en la parte central está nuestro tablero Kanban que es la visualización que hemos elegido.

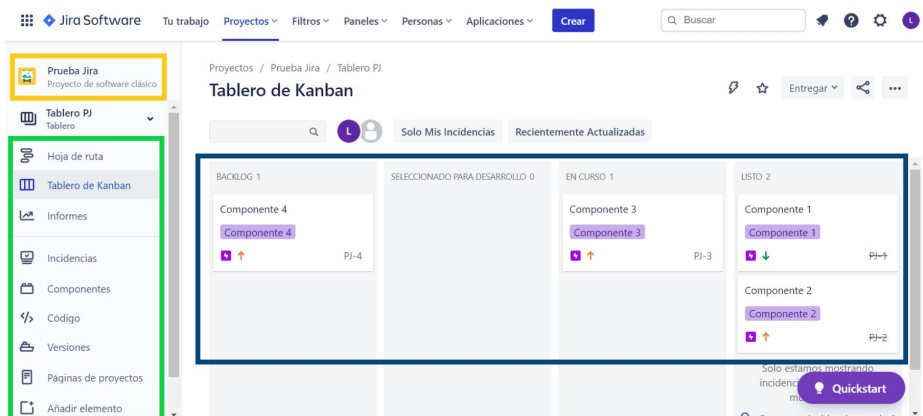


Figura 3.3: Pantalla principal de Jira Software

Contamos además con la integración de otras aplicaciones, para que podamos personalizar el proyecto a nuestra manera y nos facilite la planificación de este.

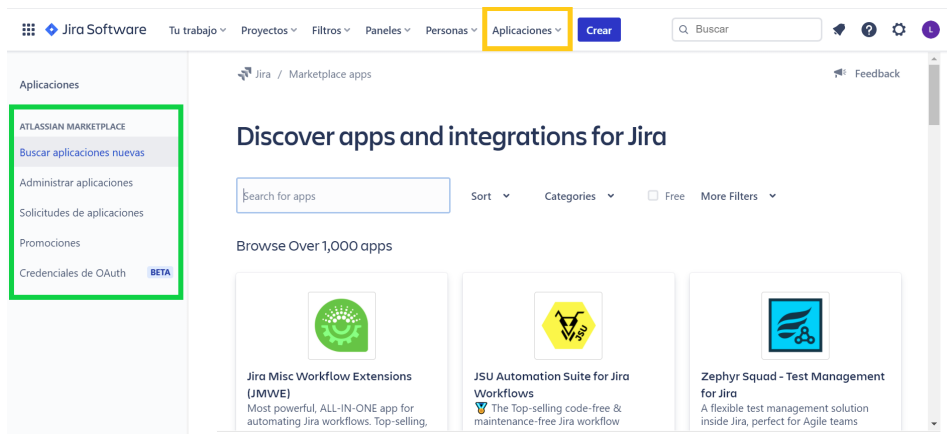


Figura 3.4: Integración con otras herramientas

Jira a parte de ser una herramienta que proporciona una gestión ágil de los proyectos también fue ideado para el seguimiento de errores y gestión de incidencias. Esto nos ofrece una visualización completa del proyecto, tanto de la parte bien desarrollada como de la que no lo está tanto. Con esto conseguimos centrarnos también en arreglar estos errores ya que, hay que solucionarlos para que el proyecto funcione.

3.2 Trello

Trello es otra de las herramientas que destaca entre las metodologías ágiles. Está basada específicamente en Kanban. Trello se caracteriza por el uso de un tablero junto con sus tarjetas movibles. Esta herramienta salió a la luz en 2011 y más tarde fue adquirida por Atlassian. [35]

Características principales:

- **Precio:** cuenta con una prueba gratuita, pero dependiendo de las funcionalidades que se quieran y el número de usuarios que haya en el equipo el precio varía. Ofrece tres planes: Free Trello que es gratuito, Business Class para equipos de hasta 100 personas con un coste de 12,5 dólares usuario/mes, por último, el Enterprise que está disponible para más de 100 personas y el precio hay que consultarlo con el equipo de Trello.
- **Disponibilidad:** se ofrece para móvil, pc o Ipad. Disponible en Windows, Android e IOS.
- **Formación:** ofrecen las guías de utilización de Trello en forma de vídeos y seminarios web, así como documentación.

Funcionalidades:

La herramienta Trello dispone de un tablero que ofrece cuatro funcionalidades básicas con infinitas adaptaciones.

1. **Tablero:** Trello no sólo es una herramienta de trabajo sino que sirve también para gestionar cualquier plan que tengas en mente, así como un viaje con amigos. Se trata de una tablero donde se muestra el seguimiento del proyecto/plan.
2. **Listas:** mediante estas listas mantenemos la organización de las tarjetas según el estado en el que se encuentre la tarea que queremos realizar. Normalmente un proyecto suele dividirse en tres listas: 'To do' , 'Doing' , 'Done'. Sin embargo, Trello es una herramienta flexible que permite la creación de tantas listas como creamos necesarias para la buena gestión de nuestro proyecto. Lo importante es que todas las tareas lleguen a su fin.
3. **Tarjetas:** son el elemento principal del tablero, ya que es donde se detallan las tareas del proyecto. En estas tarjetas se pueden añadir la descripción de la tarea, su fecha de vencimiento, el responsable de esa tarea e incluso adjuntar documentos. Durante el desarrollo del proyecto las tarjetas se van moviendo a través de las listas que se han creado para saber el estado en el que se encuentran.
4. **Menú:** el menú de Trello es la principal fuente de organización del proyecto, ya que desde ahí se añaden los miembros del equipo, se filtran las tarjetas, se cambia el fondo del tablero, se configuran los power-ups que sirven para añadir funcionalidades o para conectar Trello con otras aplicaciones como el calendario o Google drive y, por último, ver toda la actividad que se está produciendo en el proyecto.

Interfaz:

La interfaz de Trello es muy intuitiva, ya que el tablero se basa en Kanban y puedes añadir tantas columnas como necesites. Además, te permite utilizar plantillas para que no tengas que crearlo desde cero. La pantalla principal del proyecto se muestra tal y como vemos en la imagen siguiente.

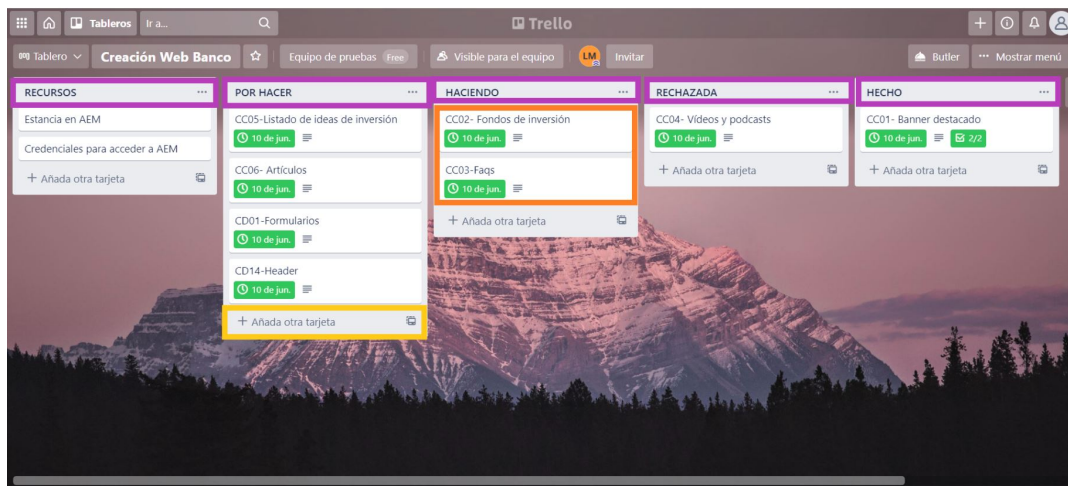


Figura 3.5: Pantalla principal de Trello

Como podemos observar, lo hemos basado en 4 columnas y cada una con las tareas que le corresponden debido a su estado. En cada tarea se pueden especificar la fecha de vencimiento, también podemos añadir una breve descripción sobre lo que trata y crear una 'checklist' en la que añadimos subtareas para ir marcando cuales de ellas se han realizado ya e ir controlando que es lo que queda para completar dicha tarea. Además, se va mostrando la actividad que se va haciendo sobre esa tarea, por ejemplo, quién ha adjuntado un documento o quién la ha movido de una columna a otra.

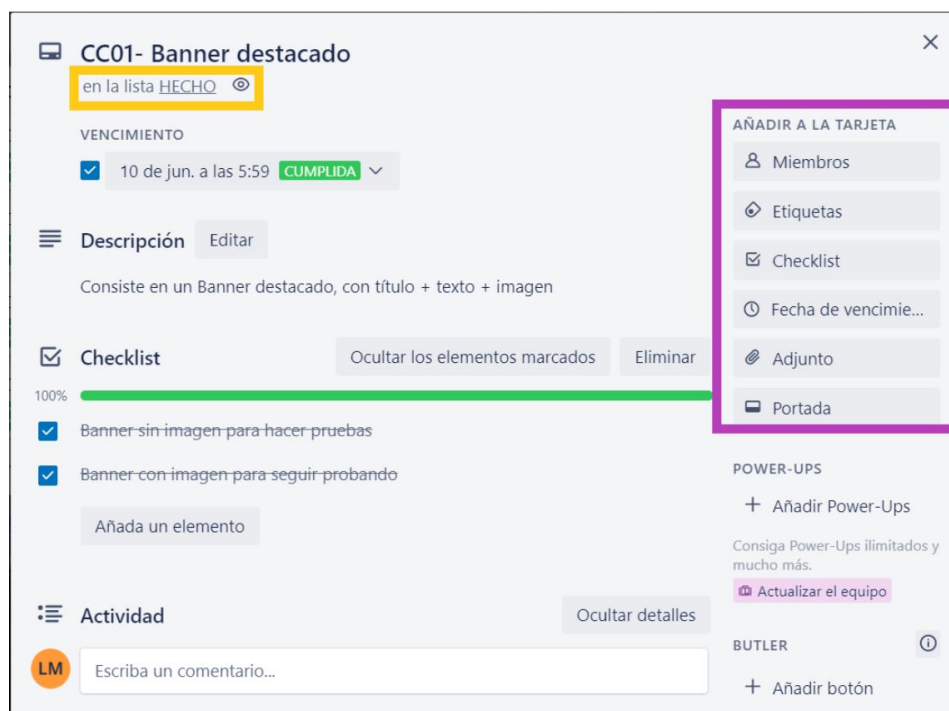


Figura 3.6: Detalle tarjetas de Trello

Trello consta de un menú en la parte derecha del tablero, en el que se puede hacer cualquier modificación del tablero. Mediante la opción 'Butler' se automatizan las tarjetas. Desde el menú podemos hacer el seguimiento del proyecto y añadir etiquetas o pegatinas.

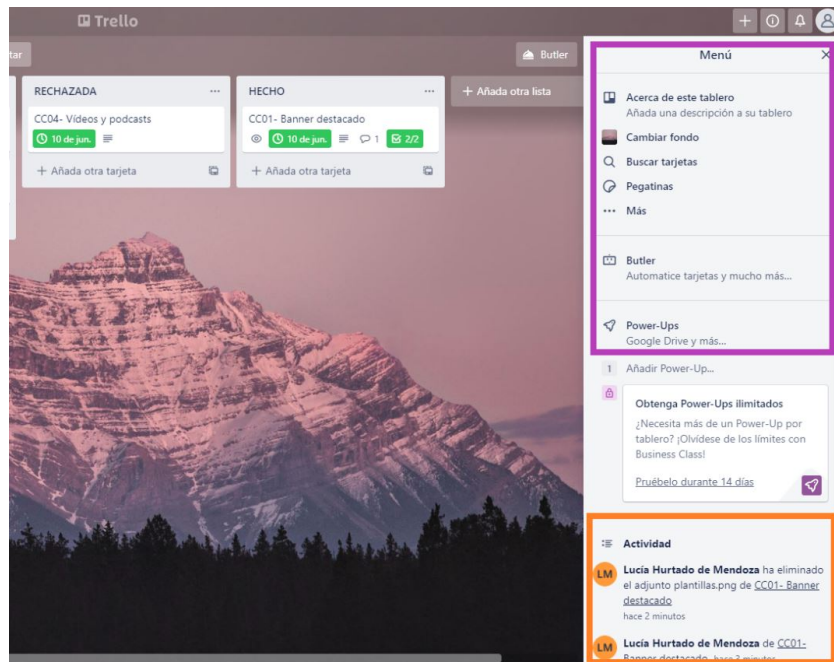


Figura 3.7: Menú lateral de Trello

También tenemos la opción de 'power-ups' que son las integraciones con otras herramientas que nos ayudan a que el proyecto sea más llevadero y sea totalmente personalizado.

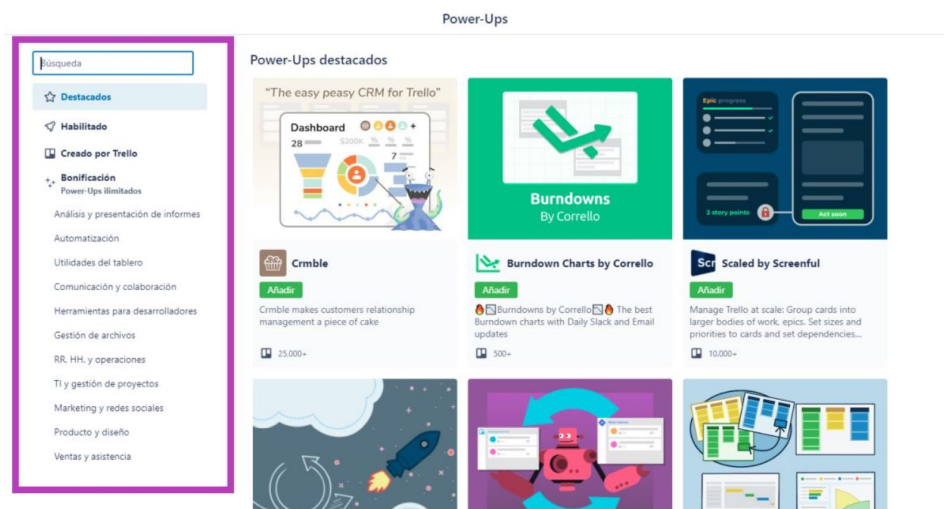


Figura 3.8: Power-ups de Trello

3.3 Wrike

Wrike trata de un software diseñado para la planificación de proyectos y coordinación de los equipos de cualquier tamaño. Fue fundada [\[16\]](#) en 2006 por Andrew Filev y se lanzó la primera versión en 2007.

Características principales:

- **Precio:** Ofrece cuatro tipos de planes en base a los usuarios que vayan a utilizar esta plataforma y las integraciones o funcionalidades que se quieran utilizar. Ofrece una prueba gratuita para cada uno de estos planes durante 14 días. El primer plan es Wrike gratuito que está disponible para 5 usuarios. Wrike profesional es de 9,80 dólares por usuario/mes para 5, 10 o 15 usuarios. Wrike negocio es de 24,80 dólares por usuario/mes desde 5 hasta 200 usuarios. Wrike empresa sin límite de usuarios y cuyo precio es pactado con Wrike.
- **Disponibilidad:** se ofrece para móvil, pc o Ipad. Disponible en Windows, Android e IOs.
- **Formación:** puedes recibir información sobre el funcionamiento de Wrike a través de webinars, guías, youtube o cursos.

Funcionalidades:

A continuación nombramos algunas funcionalidades que destacan en Wrike para conocer mejor la herramienta y saber si es la que estábamos buscando.

1. **Tablero Kanban:** gracias a estos tableros podemos entender rápidamente el estado del proyecto y las tareas, ya que cada una estará en la columna que le pertenezca, pudiendo mejorar así la eficiencia del equipo.
2. **Gestión del trabajo:** permite la creación de las tareas necesarias incluyendo las fechas de inicio y fin, el responsable de la tarea, la descripción de esta si fuera necesario y las dependencias entre ellas. Sin olvidarnos de que disponemos del diagrama de Gantt que sirve de gran ayuda para controlar del proyecto.
3. **Formularios:** personaliza los formularios de solicitud en los que se detallen los campos necesarios para la creación automática de los proyectos o tareas.
4. **Integraciones y automatización:** Wrike permite la integración con otras aplicaciones y automatiza los procesos para mejorar el rendimiento de estos y aumentar su productividad.
5. **Calendarios de Wrike:** se trata de una visualización completa del proyecto en todas las áreas para facilitar la planificación.

Interfaz:

Para analizar Wrike hemos utilizado la versión gratuita porque ofrece todas las funcionalidades que necesitamos para gestionar o controlar un proyecto .

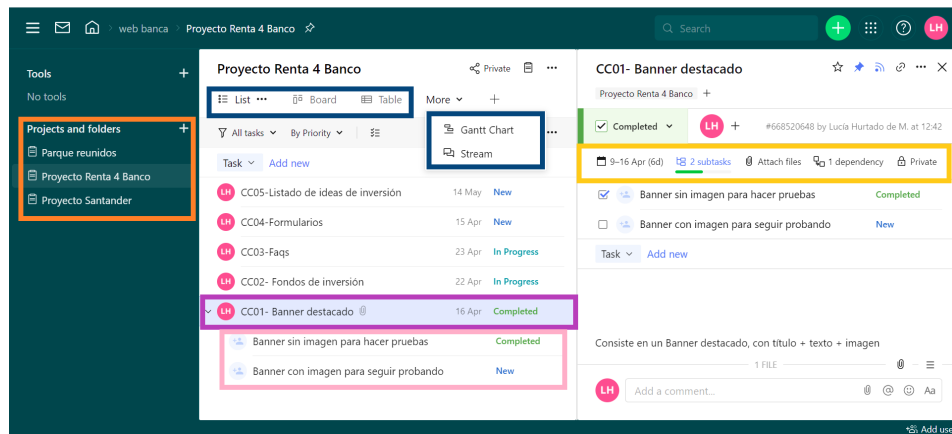


Figura 3.9: Página principal de Wrike

Esta es la pantalla principal de Wrike. Como podemos observar estamos en la visualización lista, pero en la parte superior podemos elegir también la visualización Kanban o el formato tabla. La visualización lista nos ayuda a ver los detalles de la tarea, así como su descripción, las dependencias, el estado de cada tarea (nuevo, en progreso o completado), las subtareas, la fecha de vencimiento y el miembro que la tiene asignada. En este caso en todos es LH. También vemos en el menú de la izquierda que desde la misma pantalla podemos tener controlados todos los proyectos en los que estamos asignados.

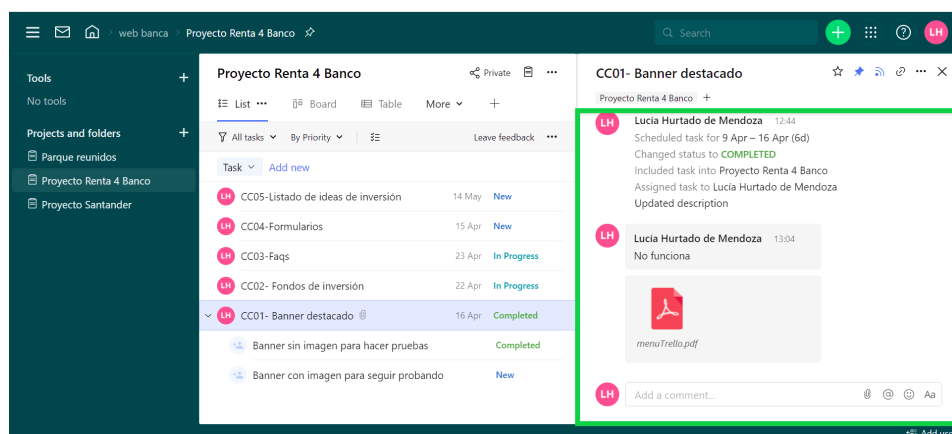


Figura 3.10: Colaboración de Wrike

Una de las cosas que nos llama la atención sobre Wrike es que ofrece soluciones colaborativas, permitiendo compartir en tiempo real archivos, enviar mensajes y mencionar a los miembros del equipo. Permite la exportación de datos a pdf, quedando la

información del proyecto totalmente resumido y a disposición de cualquier persona que lo necesite.

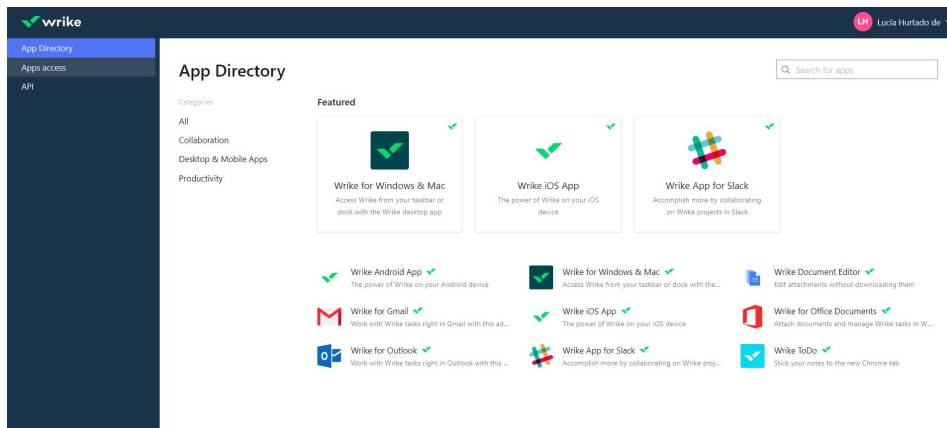


Figura 3.11: Integraciones de Wrike con otras herramientas

Como no, también nos ofrecen gran cantidad de integraciones con otras aplicaciones. Al ser la versión gratuita la que estamos analizando, el abanico de posibilidades que nos ofrece no es muy extenso. Sin embargo, en las otras herramientas que hemos analizado nos muestran las posibilidades que hay de integrar esa determinada plataforma con otras, aunque para ello haya que actualizar el plan.

3.4 Asana

Asana permite la planificación de tu proyecto destacando por la sincronización continua entre los miembros del equipo, lo que hace que sea más efectiva a la hora de buscar buenos resultados. Esta herramienta se fundó [15] en 2008 por Justin Rosenstein, pero fue accesible a usuarios por primera vez en 2011.

Características principales:

- **Precio:** Ofrece cuatro tipos de planes según las funcionalidades de las que se quiera disponer. El Basic, es el gratuito. El Premium es de 10,99 euros al mes por usuario. El Business es de 24,99 euros. El último plan es el Enterprise en el que el precio es consensuado con el equipo de Asana directamente.
- **Disponibilidad:** se ofrece para móvil, pc o Ipad. Disponible en Windows, Android e IOs.
- **Formación:** puedes recibir información sobre el funcionamiento de Asana a través de webinars, guías, youtube o cursos.

Funcionalidades:

Asana es una herramienta sencilla que permite gestionar proyectos independientemente de la complejidad o del tamaño de este.

1. **Vistas:** ofrece distintas visualizaciones para que al usuario le sea más fácil planificarse. Dispone de tableros, listas, cronograma y calendario.
2. **Organización:** Asana facilita la gestión del proyecto ofreciendo a los usuarios funcionalidades para que no pierdan el hilo del proyecto en ningún momento. Algunos de ellos son: detallar tareas y subtareas, establecer logros, adjuntar archivos, indicar las fechas de inicio y fin, establecer dependencias de tareas para que no empiece una hasta que la anterior haya finalizado, utilización de plantillas y creación de reglas para que se completen todas las acciones sin saltarse ningún paso.
3. **Informes:** en los informes podemos observar el estado de cada proyecto del que formamos parte y de las tareas que tenemos asignadas.
4. **Controles:** el jefe de proyecto es el que decide que tipo de accesibilidad y privilegios tendrá cada miembro del equipo en el proyecto.
5. **Comunicación:** hemos hablado muchas veces de lo importante que es comunicarse en un proyecto. Es por eso que Asana permite crear conversaciones entre los miembros del equipo y comentar en las tareas mencionando a la persona indicada, para que sea un proyecto claro y transparente.

Interfaz:

Asana a simple vista es una interfaz muy intuitiva y bastante más sencilla. Al principio puede darnos la sensación de que sus funcionalidades en el plan gratuito son más limitadas, pero nos ofrece todo lo que necesitamos.

En la pantalla principal nos muestra el tablero Kanban, o la visualización que hayamos elegido, tanto en resumen como lista, tablero o cronograma. En el botón de 'más' también nos ofrece más vistas como es la del calendario y la opción de mensajería para mantener conversaciones con los integrantes del proyecto. A la derecha tenemos opciones de filtrado de las tareas o de ordenación según la prioridad, fecha de vencimiento, orden alfabético... lo que más nos favorezca a cada uno. Y, como en todas las herramientas, podemos personalizarla añadiendo nuevos campos o integraciones con otras aplicaciones.

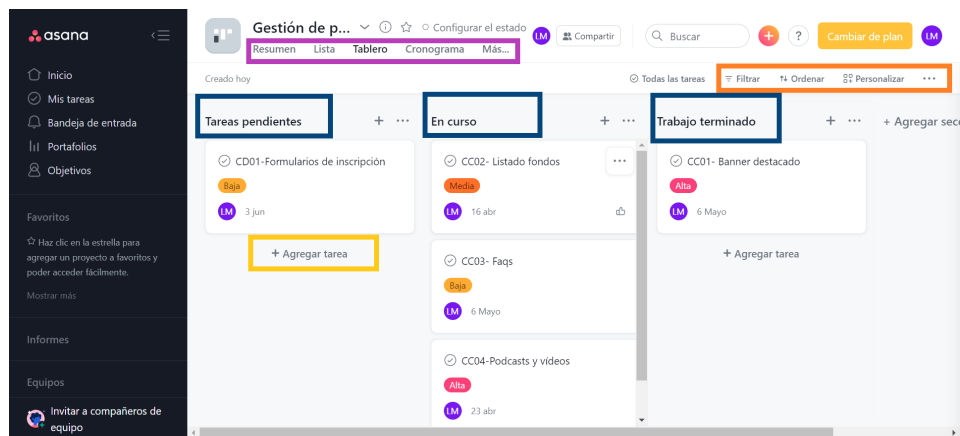


Figura 3.12: Página principal de Asana

En cuanto a la gestión de tareas, se hace a través de tarjetas que se colocan en el tablero Kanban que hemos creado. Cada tarjeta consta de un título y una descripción. Tenemos la opción de rellenar el resto de campos que vemos en la imagen para un detalle más completo de la tarea. Podemos visualizar toda la actividad de la tarjeta, sabiendo quien la ha modificado o cuándo se ha movido de estado. Como en las anteriores herramientas, se añaden subtareas y adjuntan documentos, además de otras funcionalidades que mostramos a continuación en la siguiente imagen.

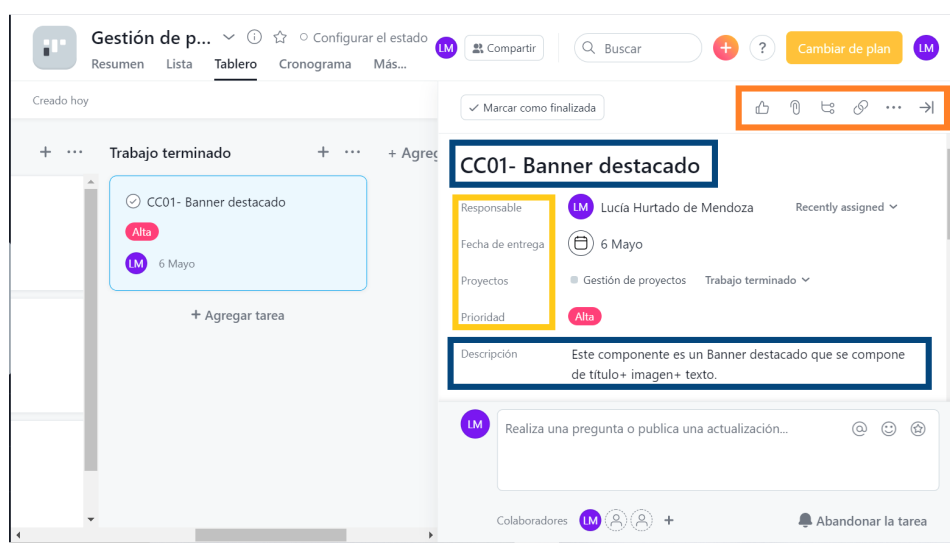


Figura 3.13: Detalle de la Tarjeta de Asana

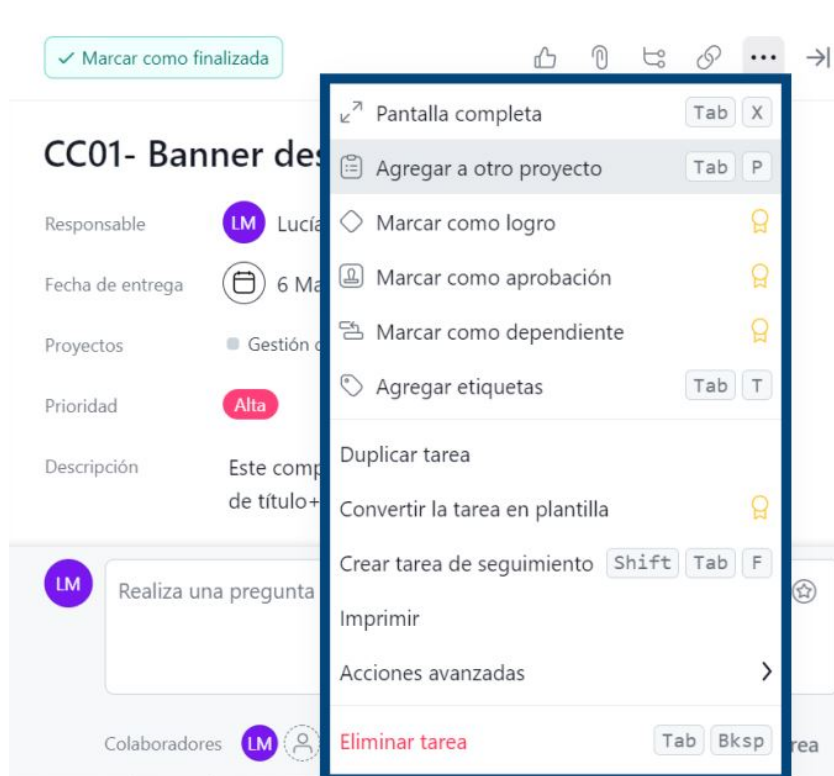


Figura 3.14: Opciones de detalle de la Tarjeta de Asana

3.5 Monday.com

Monday.com es un software de planificación de proyectos que, gracias a su flexibilidad, se adapta al tamaño de cualquier equipo, por lo que abarca tanto Start ups como grandes empresas. Esta herramienta fue fundada por Roy Mann y Eran Zinman en 2014. Fue fundada con el nombre de 'Dapulse', que más tarde pasó a llamarse 'Monday.com' porque el nombre inicial no les convencía, entonces decidieron llamarle Monday.com, ya que el comienzo de la jornada laboral son los Lunes. El principal objetivo de la creación de esta herramienta era que los miembros del equipo estuviesen encantados de usarla en su día a día porque según Roy Mann [28], el valor de la empresa es un paso más en el camino, pero lo que verdaderamente nos hace levantarnos por la mañana son las opiniones de las personas que encuentran su trabajo significativo en su día a día.

Características principales:

- **Precio:** Ofrece cuatro tipos de planes en base a las integraciones o funcionalidades que se quieran utilizar. Cada plan es mínimo para un equipo de 3 usuarios. Ofrece una prueba gratuita en el plan Pro durante 14 días. El primer plan es el Básico siendo 8 euros usuario/mes. El plan Estándar es de 10 euros por usuario/mes. El plan Pro es de 16 euros por usuario/mes. Por último, el plan Corporativo cuyo precio es pactado con Monday.com.

- **Disponibilidad:** se ofrece para móvil pc o Ipad. Disponible en Windows, Android e IOs. También está disponible para Linux en pc.
- **Formación:** puedes recibir información sobre el funcionamiento de Monday.com a través de webinars, guías, youtube o cursos.

Funcionalidades:

Nombramos algunas de las funcionalidades generales del producto para conocerlo mejor.

1. **Vistas:** permite personalizar los flujos de trabajo incorporando el número de columnas necesarias o la utilización de plantillas que más te gusten. Además, las actualizaciones son en tiempo real, notifica a cada miembro del equipo y permite agregar archivos. Ofrece también visualizaciones como Kanban, calendario, Gantt, cronograma y muchas más cosas que simplifican tu trabajo con un simple vistazo.
2. **Automatizaciones:** deja en manos de los robots las tareas que se repiten día a día en tu proyecto.
3. **Integraciones:** permite la conexión de tus datos con otras herramientas. Además, Monday.com ofrece el 'Mercado apps de Monday' que contiene aplicaciones que te ayudan a mejorar el trabajo en equipo.
4. **Paneles:** Datos del proyecto plasmados en gráficos o diagramas que nos facilitan el conocimiento del estado del proyecto con una simple ojeada.
5. **Comunicación:** como ya sabemos la comunicación es esencial por lo que Monday.com permite contactar con los compañeros a través de la herramienta sin necesidad de acceder al correo electrónico, siendo así mucho más ágil cualquier proceso.

Interfaz:

Monday.com es la última herramienta a analizar, pero no por ello la peor... Y a continuación vamos a investigar como funciona y las opciones que ofrece para la gestión de proyectos.

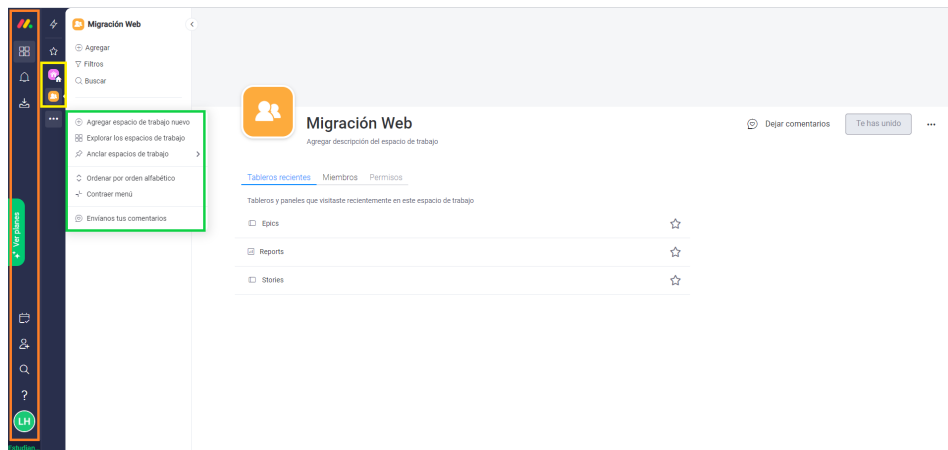


Figura 3.15: Página principal de Monday

En la pantalla principal de Monday nos encontramos el menú lateral izquierdo donde podemos consultar en este orden las siguientes opciones: los espacios de trabajo que tenemos creados, las notificaciones, el buzón, el calendario con la planificación de nuestra semana. Desde el menú podemos invitar a miembros y ver nuestro perfil. Los iconos que hemos resaltado son los dos espacios de trabajo que tenemos creados actualmente.

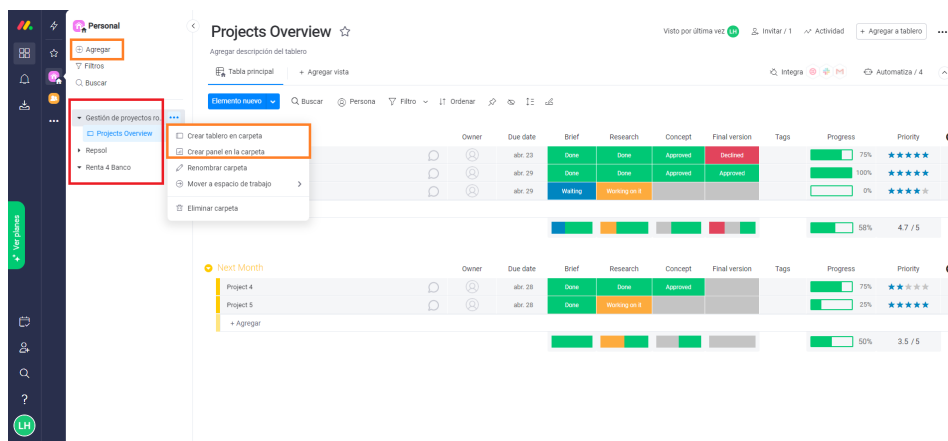


Figura 3.16: Funcionalidades dentro del Espacio de trabajo

Cuando ya nos hemos creado nuestro espacio de trabajo, desde la opción agregar podemos crear carpetas para organizar los diferentes proyectos que tengamos y se nos irán acumulando en el lateral izquierdo que hemos señalado. Una vez creados los proyectos, tenemos la opción de crear tableros para empezar con las tareas de este. Además, Monday nos ofrece la opción de utilizar plantillas.

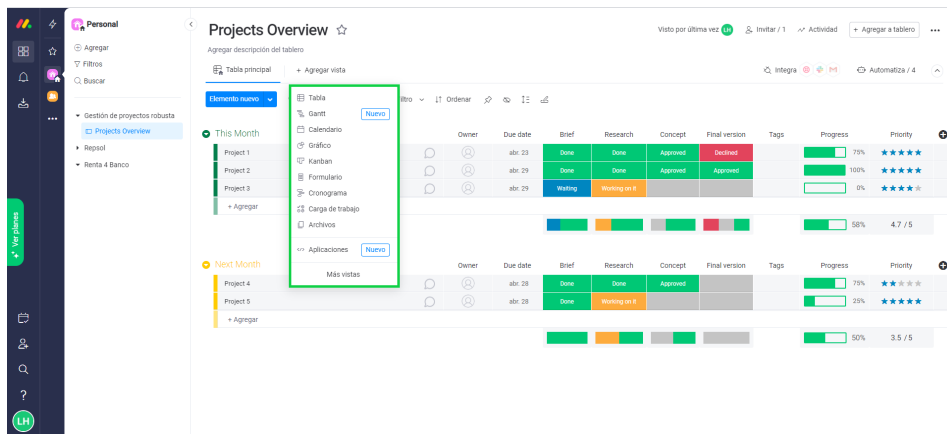


Figura 3.17: Diferentes visualizaciones de la planificación del proyecto

Esta aplicación a mi parecer destacaría por la cantidad de visualizaciones que ofrece la versión gratuita. Es una interfaz sencilla que muestra la planificación del proyecto en la vista que más guste el usuario.

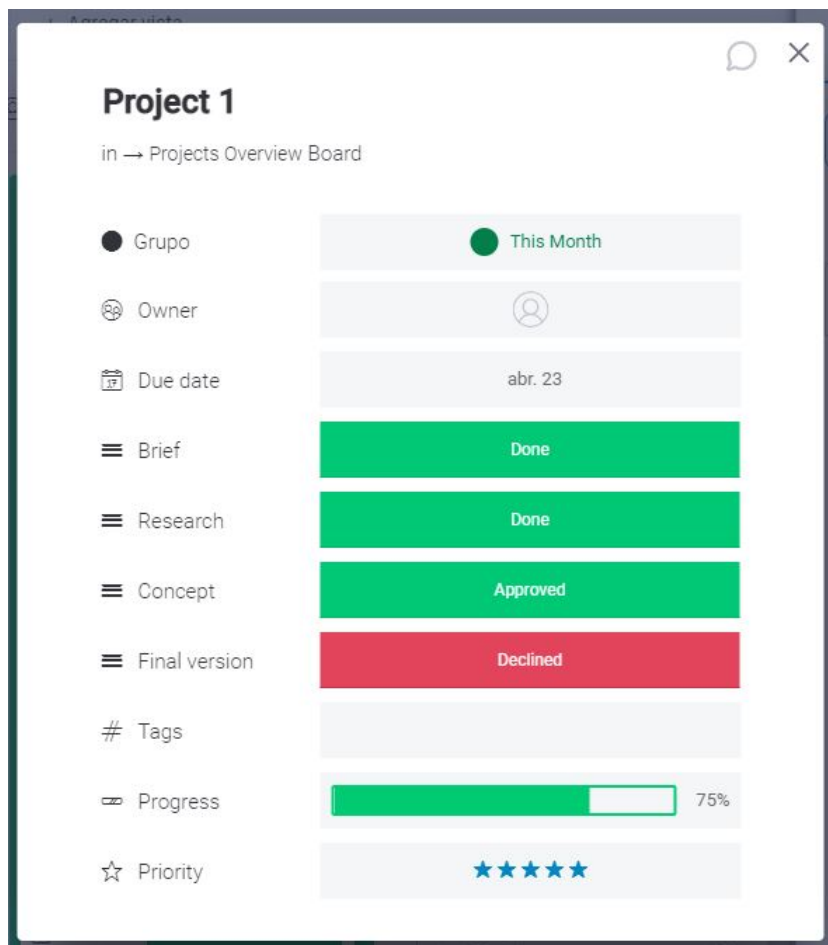


Figura 3.18: Detalle actividad de Monday

Como en todas las herramientas se puede visualizar el detalle de cada tarea o elemento creado, así como modificarlas en la ventana modal que se levanta.

3.6 Tabla comparativa de las herramientas estudiadas

Las aplicaciones que hemos visto anteriormente son algunas que existen hoy en día, pero gracias a la de información que existe ya acerca de las metodologías ágiles y sus herramientas hemos podido centrarnos en las cinco que son más conocidas o utilizadas y que mejor calificaciones o reseñas tienen por usuarios reales. Además de este breve análisis, algo que facilita mucho la elección de una herramienta es compararla con otras. Por eso creemos necesario mostrar una tabla con las herramientas que hemos desarrollado junto con una serie de funcionalidades importantes para la buena planificación de un proyecto. En esta tabla marcaremos cuáles son las funcionalidades que ofrece cada una de las herramientas, así como las funcionalidades de las que carecen. Mediante un simple vistazo podremos guiarnos por la que más nos conviene. Hemos considerado práctico realizar dos tablas: en la primera la comparación de las herramientas se hace sobre la versión gratuita, y en la segunda sobre la versión más completa de cada herramienta.

| Funcionalidades | Herramientas | | | | |
|--------------------------------------|--------------|--------|-------|-------|------------|
| | Jira | Trello | Wrike | Asana | Monday.com |
| Menciones a los miembros del equipo | ✓ | ✓ | ✓ | ✓ | ✓ |
| Comentarios/notas | ✓ | ✓ | ✓ | ✓ | ✓ |
| Controles o permisos de acceso | ✗ | ✗ | ✗ | ✗ | ✗ |
| Creación de diagramas | ✓ | ✗ | ✗ | ✓ | ✓ |
| Creación de informes y estadísticas | ✓ | ✗ | ✗ | ✗ | ✓ |
| Etiquetas | ✗ | ✓ | ✓ | ✓ | ✓ |
| Gestión de archivos/recursos | ✓ | ✓ | ✓ | ✗ | ✓ |
| Gestión de calendario | ✓ | ✗ | ✗ | ✗ | ✗ |
| Gestión de correo electrónico | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de presupuestos | ✗ | ✓ | ✓ | ✓ | ✓ |
| Gestión de proyectos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de tareas | ✓ | ✓ | ✓ | ✓ | ✓ |
| Importación y exportación de datos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Informes personalizables | ✓ | ✗ | ✗ | ✗ | ✓ |
| Integraciones con otras herramientas | ✓ | ✓ | ✓ | ✗ | ✗ |
| Plantillas de proyecto | ✓ | ✓ | ✓ | ✓ | ✓ |
| Seguimiento de fechas de vencimiento | ✓ | ✓ | ✓ | ✓ | ✓ |
| Seguimiento del proyecto | ✓ | ✓ | ✓ | ✓ | ✓ |
| Varios proyectos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Diagrama de Gantt | ✗ | ✗ | ✗ | ✗ | ✗ |
| Conversaciones por chat | ✗ | ✗ | ✓ | ✓ | ✓ |
| Actualizaciones en tiempo real | ✗ | ✓ | ✓ | ✓ | ✓ |

Figura 3.19: Comparación de las funcionalidades de las herramientas en la versión gratuita

| Funcionalidades | Herramientas | | | | |
|--------------------------------------|--------------|--------|-------|-------|------------|
| | Jira | Trello | Wrike | Asana | Monday.com |
| Menciones a los miembros del equipo | ✓ | ✓ | ✓ | ✓ | ✓ |
| Comentarios/notas | ✓ | ✓ | ✓ | ✓ | ✓ |
| Controles o permisos de acceso | ✓ | ✓ | ✓ | ✓ | ✓ |
| Creación de diagramas | ✓ | ✗ | ✓ | ✓ | ✓ |
| Creación de informes y estadísticas | ✓ | ✗ | ✓ | ✓ | ✓ |
| Etiquetas | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de archivos/recursos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de calendario | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de correo electrónico | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de presupuestos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de proyectos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gestión de tareas | ✓ | ✓ | ✓ | ✓ | ✓ |
| Importación y exportación de datos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Informes personalizables | ✓ | ✗ | ✓ | ✓ | ✓ |
| Integraciones con otras herramientas | ✓ | ✓ | ✓ | ✓ | ✓ |
| Plantillas de proyecto | ✓ | ✓ | ✓ | ✓ | ✓ |
| Seguimiento de fechas de vencimiento | ✓ | ✓ | ✓ | ✓ | ✓ |
| Seguimiento del proyecto | ✓ | ✓ | ✓ | ✓ | ✓ |
| Varios proyectos | ✓ | ✓ | ✓ | ✓ | ✓ |
| Diagrama de Gantt | ✓ | ✓ | ✓ | ✓ | ✓ |
| Conversaciones por chat | ✗ | ✗ | ✓ | ✓ | ✓ |
| Actualizaciones en tiempo real | ✗ | ✓ | ✓ | ✓ | ✓ |

Figura 3.20: Comparación de las funcionalidades de las herramientas en la versión más completa

Teniendo en cuenta toda esta información la aplicaremos en el cuarto punto, en el que desarrollaremos un proyecto y dependiendo de lo que necesitemos y a lo que vaya enfocado nuestro proyecto elegiremos dos de las herramientas expuestas anteriormente, explicando el por qué de esta elección.

Capítulo 4

Caso Práctico

4.1 Definición del proyecto y de las herramientas

Empezamos este capítulo introduciendo primero el proyecto que se va a desarrollar para gestionar las fases de las que va a constar y las tareas de las que va a estar formado. Una vez que ya sabemos a que sector va a ir enfocado el proyecto y tener una ligera idea de lo que vamos a necesitar, elegiremos las herramientas que más puedan adaptarse a nuestras necesidades.

El proyecto que vamos a gestionar es sobre el portal web de un negocio. El negocio tiene una página web ya antigua, por lo que quieren renovarla completamente. La metodología ágil que se va a llevar a cabo para la planificación del proyecto es la metodología Scrumban, ya que nos parece buena idea mezclar las mejores prácticas de la metodología Scrum y Kanban.

Para desarrollar esta página web se requiere de una organización, por eso hay que definir unos roles en los que tengamos claro las funciones que cada uno llevará a cabo desde el inicio del proyecto. A la hora de establecer estos roles nos hemos basado en Scrum, ya que son más claros que en Kanban. Por una parte tendremos a las dos figuras principales de esta metodología ágil:

- **1 Product Owner:** es la persona que mantiene el contacto en todo momento con el cliente, tanto para informar del estado del producto como para consultar cualquier duda. Además, selecciona los elementos del backlog para cada sprint.
- **1 Scrum Master:** es el que planea los sprints, organiza las reuniones diarias (daily) para ver el estado de cada tarea y ofrece su ayuda al equipo de desarrollo.

Por otra parte contamos con **el equipo de desarrollo** está formado por un conjunto de técnicos expertos en distintos campos, y se trabajará como un equipo ágil a todos los efectos, pero las especialidades de cada uno se tendrán en cuenta en las reuniones de inicio del sprint para repartir las tareas. Los equipos tienen que ser de 3 a 9 personas, y en nuestro caso seremos 7:

- **1 analista funcional:** es la persona que se encarga de entender lo que el cliente solicita para buscar la manera más sencilla posible de llevarlo a cabo y explicarle al equipo como desarrollarlo.
- **2 desarrolladores backend:** son las personas que programan esa lógica que les ha detallado el analista funcional.
- **1 desarrollador frontend:** son las personas que desarrollan la parte que va a visualizar el usuario final, mostrando los datos que recoge de la parte backend.
- **1 arquitecto software:** es la persona que se encarga de asegurar la calidad y mantenimiento del producto, y que implementa mejoras según las necesidades del cliente.
- **1 analista web:** es la persona que recoge los datos de las acciones que se realizan en la página web.
- **1 contribuidor:** se encarga de ir revisando los componentes que se crean para la web y de configurarlos en cada página para que el site vaya cogiendo forma.

Ya tenemos proyecto y equipo formado. Lo único que nos queda es gestionar las tareas a través de las herramientas que elijamos. Las herramientas que creemos que son más afines al proyecto y que nos va a facilitar la gestión de este son **Monday y Asana**. Esta elección se debe a varios motivos: el primero y principal debido a que es la base de que todo proyecto fluya correctamente, es la comodidad que sientas al utilizar estas herramientas. Al haber tenido una primera toma de contacto con ellas en el capítulo anterior podemos decir que nos hemos sentido más cómodos con estas dos, en parte, por la interfaz tan intuitiva que tienen. Otro de los motivos que ha conducido a esta decisión es que, al final, las cinco herramientas tienen prácticamente las mismas funcionalidades y más siendo la versión gratuita la que vamos a utilizar, por lo que el nivel funcional de las herramientas no tiene gran peso en esta decisión. Un factor que hemos tenido en cuenta a la hora de la elección es un estudio que realizó FinancesOnline, que se trata de una plataforma que recopila información de las herramientas y realiza una comparación para ayudar a los usuarios a elegir la mejor. La información que tienen en cuenta a la hora de analizar las herramientas son sus características, rendimiento, integración, atención al cliente, precios y opiniones de usuarios, así conseguimos una puntuación totalmente objetiva. En el estudio [\[36\]](#) que se realizó en 2020 se generó una lista con 25 herramientas de gestión de proyectos. Entre las top 8 mejores herramientas de gestión de proyectos encontramos las que hemos analizado nosotros.

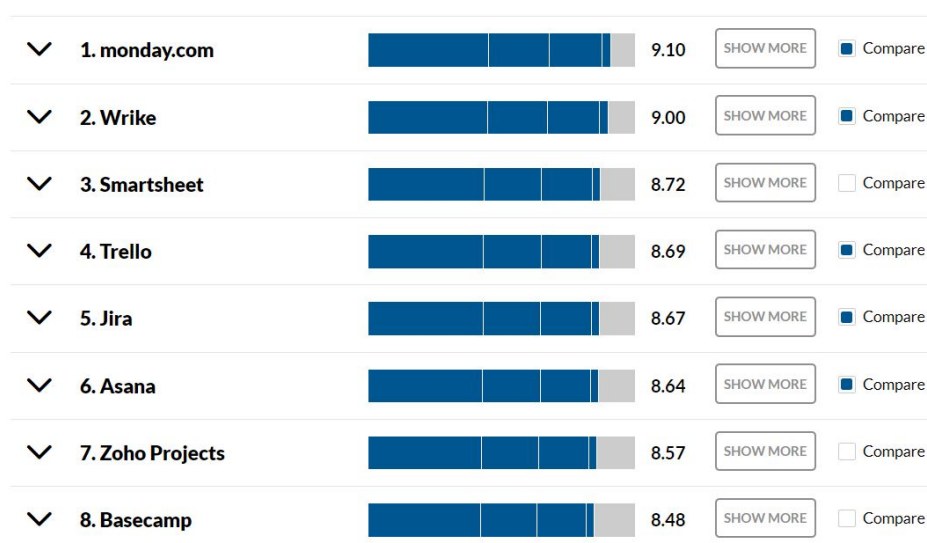


Figura 4.1: Las 8 mejores soluciones software de gestión de proyectos según Finances Online en 2020

Como podemos observar en el ranking, Monday.com se sitúa la primera. Esto nos ha llevado a elegir Monday como una de las opciones no sólo por los motivos anteriores, sino porque, personalmente la conozco menos que Jira Y Trello, de las cuales probablemente todos hayamos oído hablar de ellas y puede que conozcamos a nivel funcional un poco más que del resto. En el caso de haber elegido Asana en vez de cualquiera de las otras herramientas, todas ellas estando por encima de Asana, es porque queremos contrastar las diferencias con la primera de la lista.

Resumiendo, los motivos de la elección son más personales que técnicos debido a que las funcionalidades de las herramientas en versión gratuita son prácticamente los mismos: cumplen los requisitos para gestionar nuestro proyecto, son intuitivos, nos hemos sentido cómodos usándolas y, como dijo Winston Churchill "el éxito no es otra cosa más que brincar de fracaso en fracaso sin perder el entusiasmo", por lo que hay que arriesgar probando herramientas nuevas para saber cuál es con la que mejor trabajamos.

4.2 Plan de ejecución

En este apartado vamos a definir las fases de las que constará el proyecto, así como las tareas que son necesarias realizar. Como hemos dicho antes, la metodología ágil que se va a implementar es la Scrumban en la que mezclaremos sus mejores prácticas. Adoptamos la visualización del flujo de trabajo de Kanban en el que mostramos las tareas en las columnas 'Backlog', 'To do', 'Doing' y 'Done'. Además, contamos con el aspecto WIP de Kanban para así priorizar tareas y no dar lugar a cuellos de botella. De la metodología Scrum adoptamos la planificación por Sprints, en la que el proyecto consta de diferentes fases que hay que cumplir en el periodo de tiempo que se ha establecido con el cliente. También tendremos las reuniones diarias en las que comentamos el estado de nuestra

tarea y la actividad que se desempeñará ese mismo día, así como cualquier duda que surja. Como el proceso que se va a desarrollar va a ser el mismo en ambas herramientas, las tareas también. Al tratarse de migrar la antigua página web del negocio a una nueva tenemos que crear los componentes que vamos a utilizar, que son los siguientes:

1. **Cabecera:** La cabecera será el componente fijo y común en todas las páginas del site que estará compuesto del logo de negocio y el menú de navegación.
2. **Pie de página:** También será un componente fijo y común en todas las páginas del site. Los elementos de los que consta son el logo del negocio y los iconos enlazables a las correspondientes redes sociales.
3. **Banner destacado:** Está formado por un título (obligatorio), una descripción y un botón enlazable a una página.
4. **Tabla dinámica texto:** Este componente está formado por un título y una tabla. El título tendrá la apariencia habitual con la que se presenta en otros componentes y se mostrará con alineación centrada.
5. **Características:** Listado de características, de dos a seis elementos. Cada elemento del listado está precedido de un icono de tipo “check” y consta, a su vez, de:
 - Título, que será contribuible.
 - Texto (opcional) limitado a 3 líneas.
 - Enlace interno o externo a documento (opcional).
6. **Formulario:** Formulario para que el usuario pueda inscribirse solicitando la cita e indicando cuál es el servicio que necesita. En el formulario se proporcionarán los siguientes campos: nombre, apellidos, correo electrónico, número de teléfono y tipo de servicio que se necesita.
 - Nombre y apellidos.
 - Correo electrónico.
 - Número de teléfono.
 - Descripción donde se detalla el tipo de servicio a solicitar.

Además de las tareas contamos con los miembros del equipo y, para hacerlo más realista deberíamos crear usuarios para asignarles las tareas. Estos usuarios van a ser:

- **Nacho:** será el Product Owner.
- **Lucía:** será el Scrum Master.
- **Alberto:** se encargará de la parte funcional del proyecto y de hacer pruebas.

- **Henar y Rubén:** desarrolladores backend.
- **Marta:** desarrolladores frontend.
- **David:** arquitecto software.
- **Laura:** se encargará de la analítica.
- **Irene:** se encargará de la parte de contribución.

Nosotros partiremos directamente de la gestión interna del proyecto que nos han solicitado, es decir, no entramos en la documentación anterior que se ha elaborado para llevar a cabo el proyecto.

El proyecto va a tener una duración de 3 meses, pero esta fase durará solo 4 semanas que es lo que incluimos en la herramienta. Establecemos la fecha de inicio el 15/03/2021 y la fecha de fin de proyecto el 21/04/2021.

El primer paso para gestionar un proyecto cuando ya sabemos que es lo que el cliente necesita para su página web, es realizar un documento funcional donde se detalla el funcionamiento y la información necesaria de cada uno de los componentes que va a tener la página web. Este documento es el que necesitarán los desarrolladores backend para crear la lógica de cada componente. Una vez hecho el documento funcional los miembros del equipo pueden empezar a realizar sus tareas. El proyecto lo vamos a dividir en 4 sprints, todos dentro de la fase de construcción, que es la fase que vamos a plasmar en las herramientas. A continuación, explicamos lo que vamos a hacer dentro de cada Sprint para poder llevarlo a cabo..

Sprint 1

Al ser la primera parte del desarrollo del proyecto, la mayor parte de carga del trabajo va destinado a la creación del documento funcional, donde se explica detalladamente como tienen que funcionar los componentes que vamos a desarrollar, cuáles van a ser los campos contribuibles y cuales van a ser extraídos de aplicaciones o herramientas externas. Al margen de esto, el sprint cubrirá las historias de usuario US1, US2, US3 y US4. Además, en todos los sprints habrá una fase de revisión para comprobar que hemos cumplido con los objetivos antes de hacerle la entrega al cliente. Así podremos confirmar si está todo correctamente, solventar las incidencias que hayan aparecido o establecer mejoras.

- **US1 Desarrollo Backend de la cabecera**
- **US2 Desarrollo Frontend de la cabecera**
- **US3 Desarrollo Backend del pie de página**
- **US4 Desarrollo Frontend del pie de página**
- **US5 Desarrollo Backend del banner destacado**

- **US6 Desarrollo Fontend del banner destacado**

La duración es de 2 semanas, empezando el 15/03/2021 hasta el 26/03/2021.

Sprint 2

En esta segunda etapa de la fase de construcción, seguimos con el desarrollo y maquetación de los componentes que nos faltan.

- **US7 Desarrollo Backend de la tabla dinámica texto**
- **US8 Desarrollo Frontend de la tabla dinámica texto**
- **US9 Desarrollo Backend del módulo características**
- **US10 Desarrollo Frontend del módulo características**
- **US11 Desarrollo Backend del formulario**
- **US12 Desarrollo Frontend del formulario**

Esta fase dura 2 semanas, empezando el 29/03/2021 y acabando el 09/04/2021.

Sprint 3

Toda la parte de desarrollo ha sido finalizada, entonces comenzamos a trabajar con el contenido real de la web y con temas de analítica y SEO.

- **US13 Subir contenido del cliente**
- **US14 Posicionamiento SEO**
- **US15 Analítica**
- **US16 Configuración de los componentes contribuyendo el contenido real del cliente que se ha subido**
- **US17 Revisión completa de la página web**

La duración de este sprint es de 2 semanas, finalizando la fase de construcción el 21/04/2021.

Llegados a este punto, se hará el lanzamiento de la página web o también llamado Go live.

4.3 Monday.com

Monday.com es la primera herramienta con la que vamos a trabajar. Mostraremos como es la planificación de nuestro proyecto con Monday.com, resaltando tanto las funcionalidades que nos facilitan todo el proceso como las que echamos en falta. Antes de nada, recalcar que la versión que vamos a utilizar es el plan básico, el plan de estudiante.

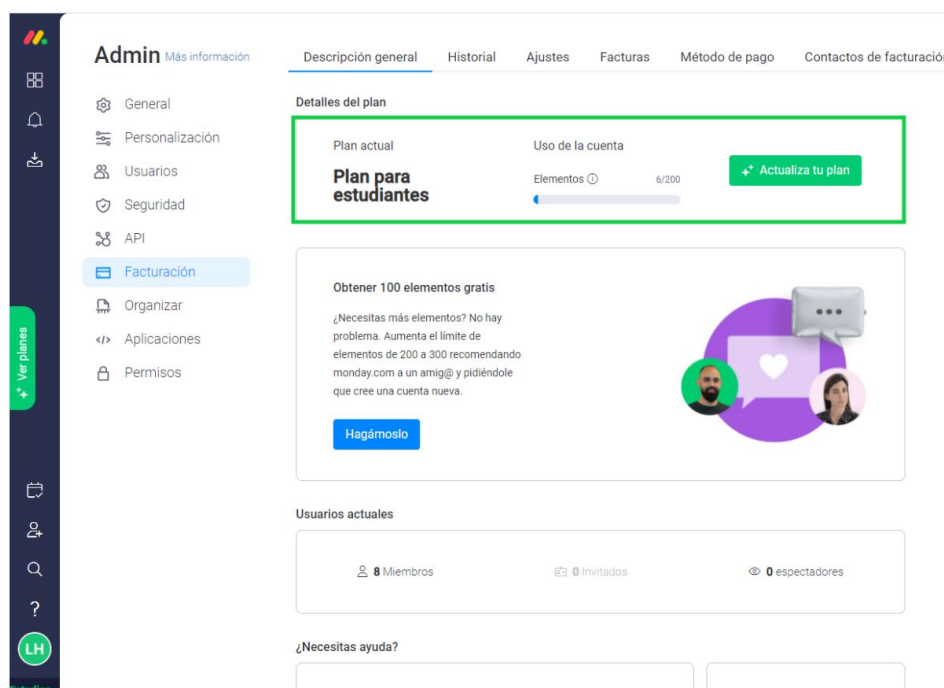


Figura 4.2: Plan para estudiantes

Gracias al análisis de las herramientas que hemos realizado en el capítulo anterior podemos saber como utilizar la herramienta. En el menú lateral izquierdo nos encontramos con los diferentes espacios de trabajo que tenemos creados. Nosotros vamos a trabajar en el que hemos llamado 'Migración Web'. Vamos a realizar la planificación de los sprints. Para ello tenemos que crearnos una carpeta y dentro de esa carpeta un tablero. Gracias a que Monday tiene una gran variedad de plantillas, que mostramos en la figura 4.4, nos es más fácil organizar el proyecto que creándolo nosotros mismos desde cero. De todas las plantillas que se nos ofrece hemos elegido la la opción 'Sprint Planning Board' de la categoría desarrollo de software porque es en lo que se va a basar nuestro proyecto.

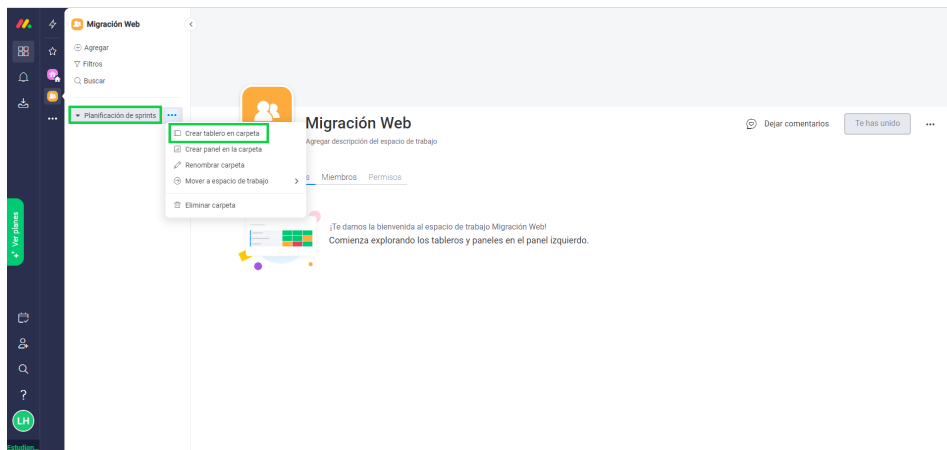


Figura 4.3: Creación de nuestro proyecto y sus tableros

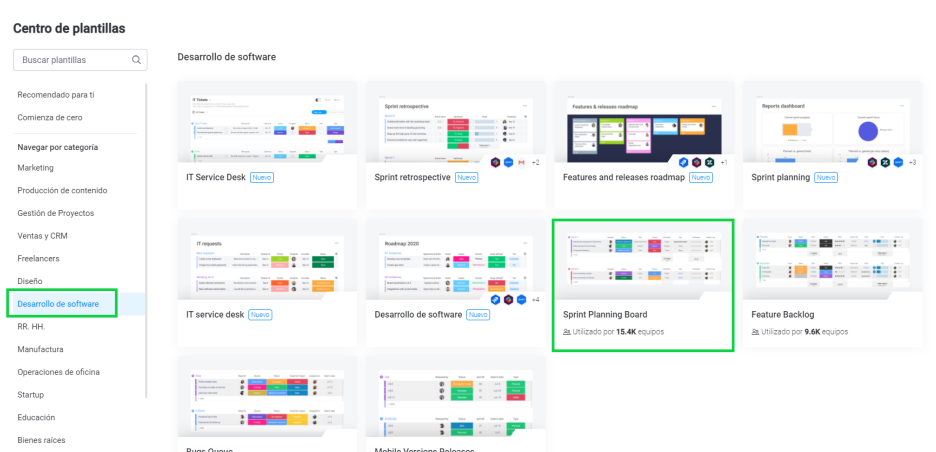


Figura 4.4: Plantillas a nuestra disposición

Una vez elegida la plantilla nosotros rellenamos los campos con los datos de nuestro proyecto, así como si queremos añadir o quitar otros campos que no están por defecto en esta plantilla. A continuación, mostramos como queda la planificación de nuestros Sprints. Hemos considerado básicas esas columnas sin necesidad de añadir ninguna más. La estimación del tiempo que se ha asignado a cada tarea se ha basado en el criterio personal del Scrum Master debido a su previa experiencia en otros proyectos y a la complejidad de cada tarea, así como las dependencias de unas tareas y otras, ya que por ejemplo hasta que el documento funcional no esté finalizado los miembros del equipo no podrían empezar a desarrollarlos porque no sabrían su debido comportamiento. También hemos querido añadir la columna archivo para adjuntar el documento funcional y que todo el equipo acceda a él.

Sprint 1

| | Asignado a | Estado | Prioridad | Estimación | Cronograma | Depende de ① | Archivo |
|--|------------|-----------|-----------|------------|--------------------|--------------------|---------|
| Documento funcional | | Haciendo | Alta | 3 days | mar. 15 - 17 | - | |
| US1 Desarrollo Backend de la cabecera | | Haciendo | Media | 2 days | mar. 18 - 19 | Documento fun... | |
| US2 Desarrollo Frontend de la cabecera | | Por hacer | Baja | 1.5 days | mar. 22 - 23 | US1 Desar... | |
| US3 Desarrollo Backend del pie de página | | Haciendo | Media | 2 days | mar. 18 - 19 | Documento fun... | |
| US4 Desarrollo Frontend del pie de página | | Por hacer | Baja | 1.5 days | mar. 23 - 24 | US3 Desar... | |
| US5 Desarrollo Backend del banner destacado | | Por hacer | Media | 2 days | mar. 22 - 23 | Documento fun... | |
| US6 Desarrollo Frontend del banner destacado | | Por hacer | Baja | 1.5 days | mar. 24 - 25 | US5 Desarrollo ... | |
| Revisión | | Por hacer | Media | 1 days | mar. 26 | Document... | |
| + Agregar | | | | | | | |
| | | | | | 14.5 days Total | mar. 15 - 26 | |

Figura 4.5: Sprint 1

Sprint 2

| | Asignado a | Estado | Prioridad | Estimación | Cronograma | Depende de ① | Archivo |
|---|------------|---------|-----------|------------|------------------|------------------|---------|
| US7 Desarrollo Backend de la tabla dinámica texto | | Backlog | Media | 3 days | mar. 29 - 31 | Documento fun... | |
| US8 Desarrollo Frontend de la tabla dinámica texto | | Backlog | Media | 2 days | abr. 1 - 2 | US7 Desar... | |
| US9 Desarrollo Backend del módulo características | | Backlog | Media | 2 days | mar. 29 - 30 | Documento fun... | |
| US10 Desarrollo Frontend del módulo características | | Backlog | Baja | 2 days | abr. 5 - 6 | US9 Desar... | |
| US11 Desarrollo Backend del módulo características | | Backlog | Media | 3 days | abr. 1 - 5 | Documento fun... | |
| US12 Desarrollo Frontend del módulo características | | Backlog | Baja | 2 days | abr. 7 - 8 | US11 Desa... | |
| Revisión | | Backlog | Media | 1 days | abr. 9 | US7 Desar... | |
| + Agregar | | | | | | | |
| | | | | | 15 days Total | mar. 29 - abr. 9 | |

Figura 4.6: Sprint 2

Sprint 3

| | Asignado a | Estado | Prioridad | Estimación | Cronograma | Depende de ① | Archivo |
|--|------------|---------|-----------|------------|--------------------|--------------------|---------|
| US13 Subir contenido del cliente | | Backlog | Alta | 1 days | abr. 12 | - | |
| US14 Posicionamiento SEO | | Backlog | Alta | 3 days | abr. 12 - 15 | - | |
| US15 Analítica | | Backlog | Alta | 5 days | abr. 12 - 16 | - | |
| US16 Configuración de los componentes contribuyendo el contenido real del cliente que se ha subido | | Backlog | Alta | 3 days | abr. 13 - 15 | US13 Subir cont... | |
| Revisión completa de la página web | | Backlog | Media | 2.5 days | abr. 19 - 21 | US13 Subi... | |
| + Agregar | | | | | | | |
| | | | | | 14.5 days Total | abr. 12 - 21 | |

Figura 4.7: Sprint 3

En el caso de querer más columnas sólo tendríamos que darle al '+' que se encuentra al final de la tabla en la esquina derecha y se nos desplegarían una gran variedad de opciones.

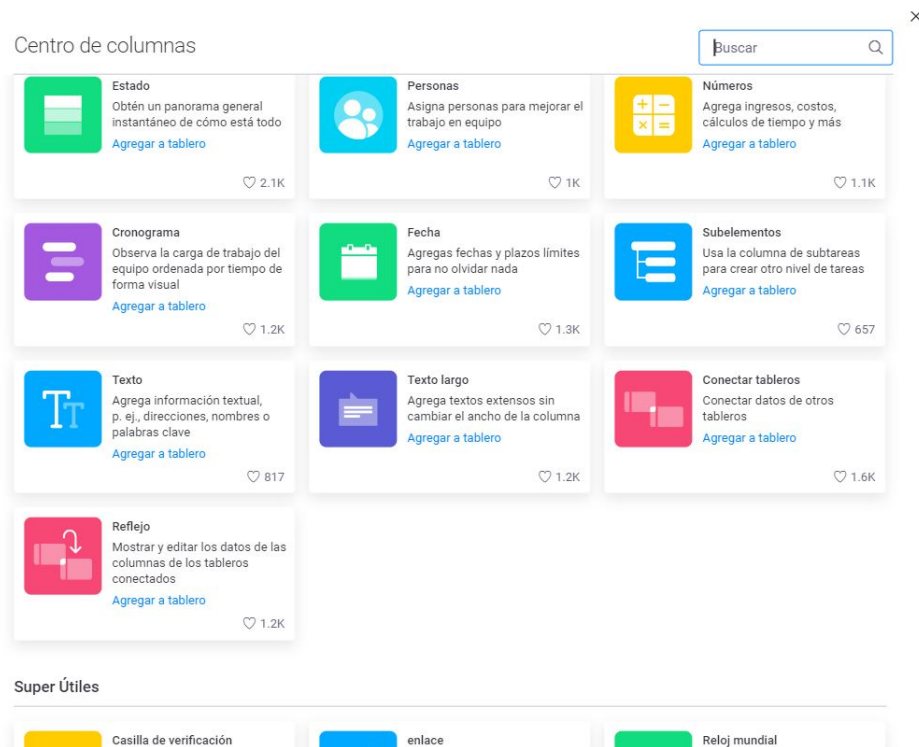


Figura 4.8: Opciones de columnas en el tablero

Como comentamos ya en el análisis de esta herramienta, una de las cosas que más llamó nuestra atención es que nos ofreciese varias vistas, ya que algunas nos guían mejor o nos parecen más sencillas o visuales. Entre ellas no podía faltar Kanban. En el menú lateral derecho tenemos la configuración de nuestro tablero Kanban y en nuestro caso hemos querido que las columnas se ordenen según el estado, pero podrían agruparse por prioridad. Disponemos de la opción de elegir la información que queremos que se visualicen en las tarjetas del tablero, mostrando así las más significativas para el seguimiento de las tareas. Por último, hemos marcado la casilla de dividir en grupos para tener bien marcado lo que se hace en cada Sprint pero, si esa casilla no la marcásemos estarían todas las actividades en el mismo tablero.

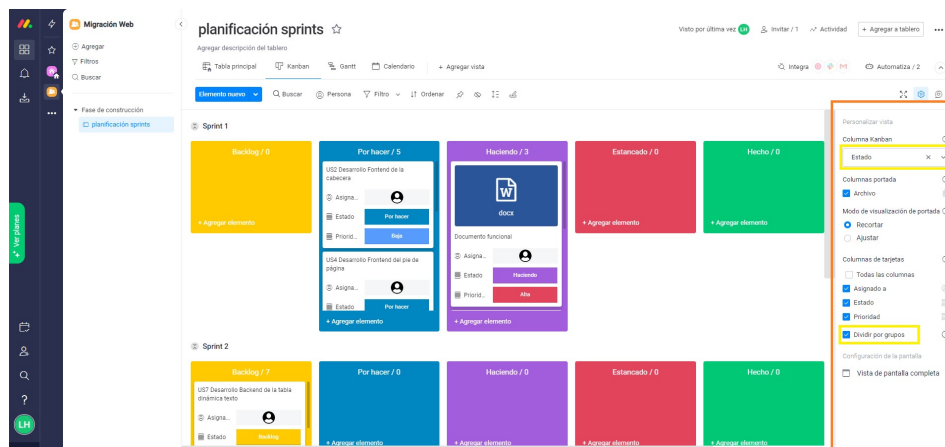


Figura 4.9: Tablero Kanban

Si queremos ver el detalle de las tarjetas más detenidamente o modificar cualquier campo sólo tenemos que pinchar en la que queramos y nos saltaría una ventana modal con sus correspondiente información.

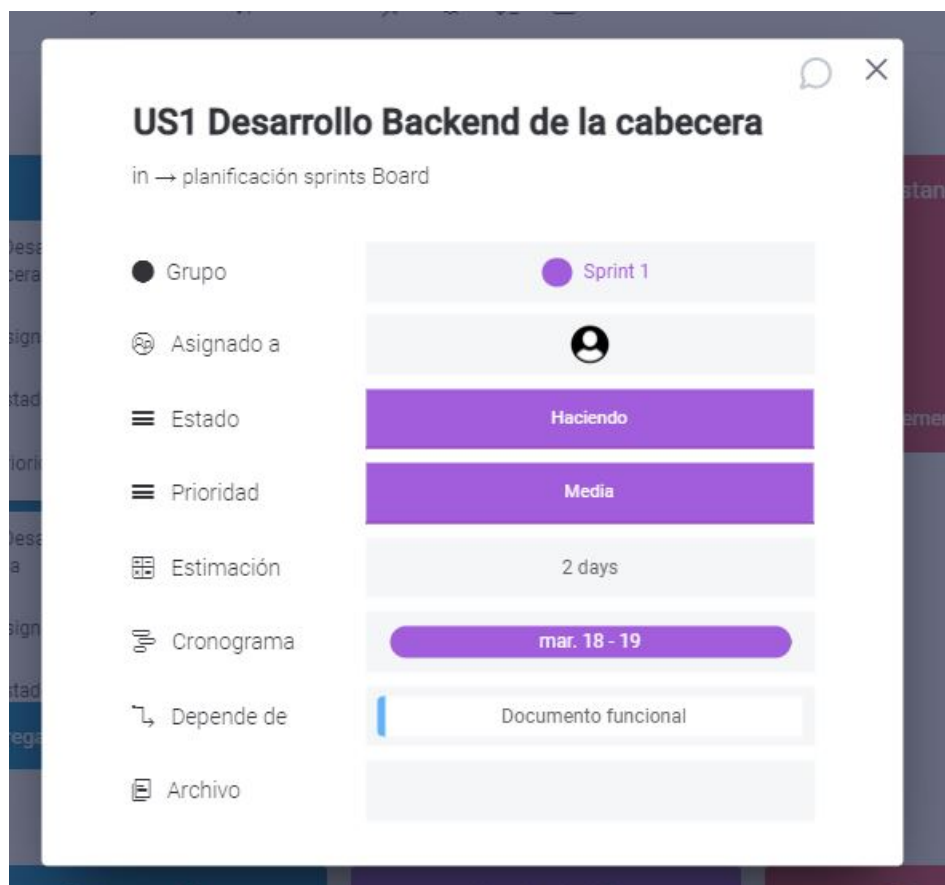


Figura 4.10: Detalle US1

Otra de las visualizaciones que es importante en cualquier gestión de un proyecto es el diagrama de Gantt, ya que podemos ver en una línea temporal dónde se sitúa cada tarea, la duración de cada sprint y la dependencia de las tareas. Algo que hemos echado en falta en esta visualización es el no poder establecer hitos, que son los puntos importantes de un proyecto, por ejemplo, nos hubiese gustado marcar como hitos el final de cada uno de los sprints para indicar que dejamos esa etapa para comenzar otra. En el caso de que queramos compartir este diagrama de Gantt, Monday nos ofrece la oportunidad de exportarlo a Excel.

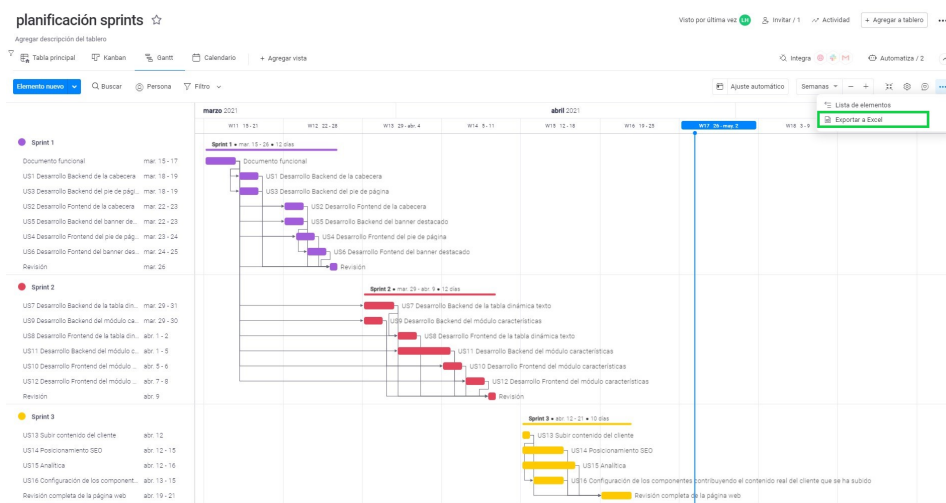


Figura 4.11: Diagrama de Gantt

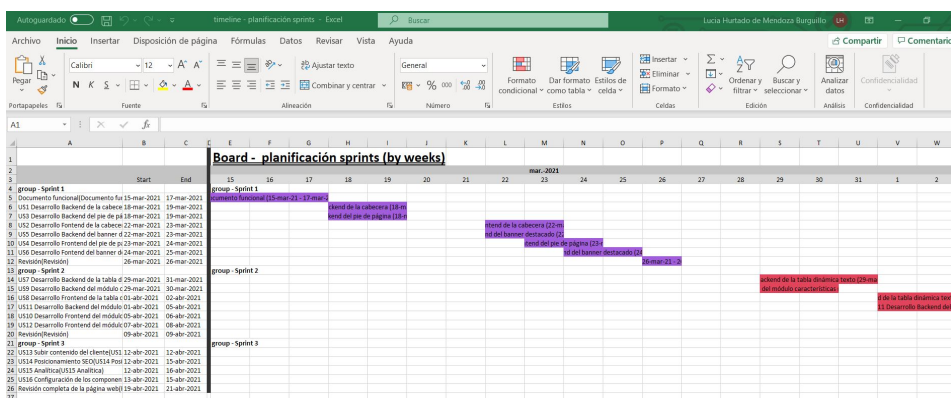


Figura 4.12: Diagrama de Gantt exportado a Excel

Hemos completado satisfactoriamente el primer Sprint, por lo que movemos algunas de las tareas del segundo sprint a la columna de 'Por hacer'. Como ya vamos más o menos por la mitad de la fase de construcción, hemos querido crear un gráfico sobre esta fase. Tenemos varios paneles para elegir como plasmar la información. También podemos combinar la información de varios tableros para crear el diagrama.

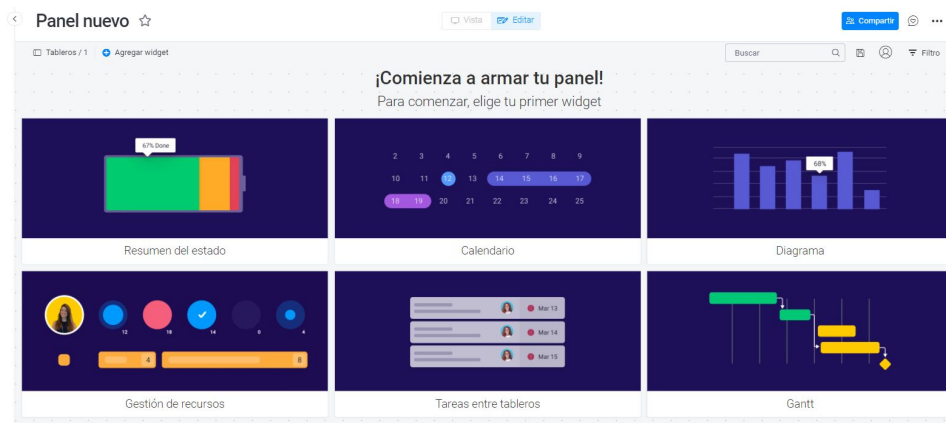


Figura 4.13: Paneles Monday

Una vez elegido el panel diagrama, nos aparece a la izquierda un menú con los tipos de diagramas que queremos elegir y la información que queremos que nos muestre. En nuestro caso hemos elegido el diagrama circular con el porcentaje del estado de las columnas en las que se encuentran las tareas.

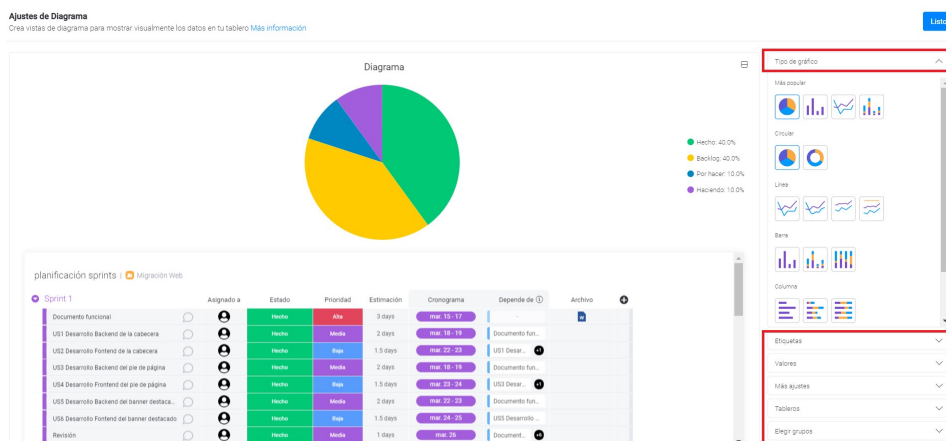


Figura 4.14: Ajustes diagrama

Cuando ya tenemos lo que queremos presionamos el botón de 'Listo' y queda tal que así:



Figura 4.15: Gráfico que representa el estado actual del proyecto

Conforme el proyecto va avanzando, los gráficos que hayamos creado se van actualizando solos.

Hemos creado otro tablero en nuestro proyecto con la plantilla de 'My Onboarding Board' para que todos los miembros del equipo estemos al tanto de las reuniones que tenemos planificadas, así como las que podamos organizar conforme el proyecto avance. Hemos considerado necesarias esas columnas ya que ofrecen la información de la reunión, el tiempo que se ha establecido es en minutos, y el número de teléfono del responsable por si se tratara de una urgencia.

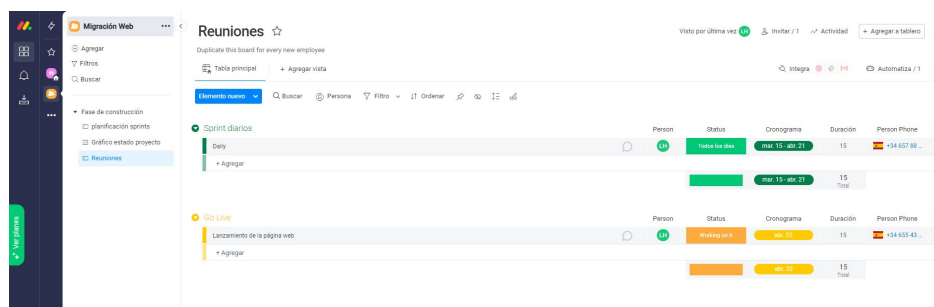
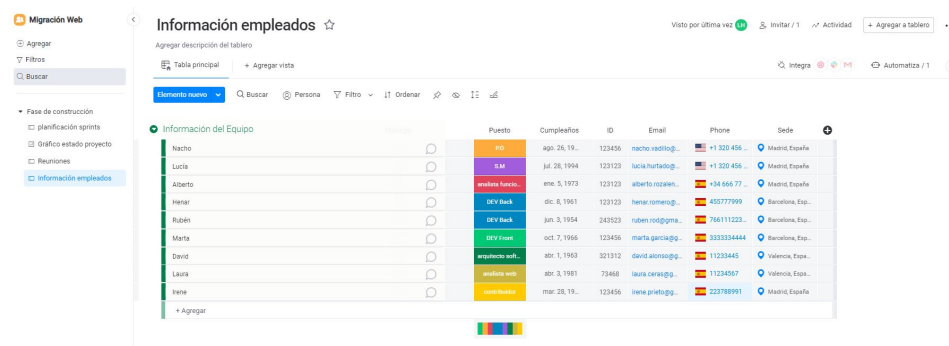


Figura 4.16: Tablero de reuniones

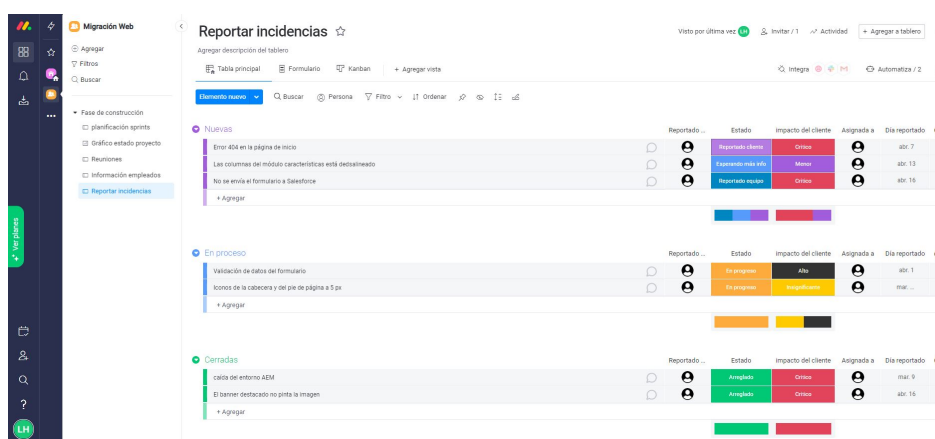
Hemos creado otra tabla con la información de cada uno de los miembros del equipo para contactar rápidamente en caso de emergencia.



| Nombre | Puesto | Cumpleaños | ID | Email | Phone | Sede |
|---------|------------------|----------------|--------|--------------------|---------------|-----------------|
| Nacho | CEO | ago. 26, 19... | 123456 | nacho.vestiblog... | +1 320 456... | Madrid, España |
| Lucía | SM | jul. 28, 1994 | 123123 | lucia.hurtadog... | +1 320 456... | Madrid, España |
| Alberto | Marketing Senior | ene. 5, 1973 | 123123 | alberto.rozaw... | +34 666 77... | Madrid, España |
| Renar | DEV Back | dic. 8, 1961 | 123123 | renar.romerog... | 45577999 | Barcelona, Esp. |
| Rubén | DEV Back | jun. 3, 1954 | 243523 | ruben.rodrigom... | 76611223... | Barcelona, Esp. |
| Marta | DEV Front | oct. 7, 1966 | 123456 | marta.parcipag... | 333334444 | Barcelona, Esp. |
| David | Marketing Mkt. | abr. 1, 1963 | 321312 | david.alonsoag... | 11233445 | Valencia, Esp. |
| Laura | Marketing Mkt. | abr. 3, 1961 | 73468 | laura.cerastag... | 11234567 | Valencia, Esp. |
| Irene | Marketing Mkt. | mar. 28, 19... | 123456 | irene.priestag... | 222789991 | Madrid, España |

Figura 4.17: Tablero con información de los empleados

Gracias a Monday tenemos la opción de crear también un tablero para reportar las incidencias con la plantilla 'Bugs queue' en el que nosotros hemos querido reportar algunas y a continuación adjuntamos ejemplo.



| Reportado por | Estado | Impacto del cliente | Asignado a | Fecha reportada |
|------------------|--------------------|---------------------|--------------|-----------------|
| Reportado por... | Reportado cliente | Crisis | Reportado... | abr. 7 |
| Reportado por... | Examinado más info | Medio | Reportado... | abr. 13 |
| Reportado por... | Reportado cliente | Crisis | Reportado... | abr. 15 |
| Reportado por... | En proceso | Alto | Reportado... | abr. 1 |
| Reportado por... | En proceso | Impacto alto | Reportado... | mar. ... |
| Reportado por... | Resuelto | Crisis | Reportado... | mar. 9 |
| Reportado por... | Resuelto | Crisis | Reportado... | abr. 15 |

Figura 4.18: Tablero de reporte de incidencias

Cuando ya hemos completado todas las tareas, se colocan las etiquetas de 'Hecho' y la fase de construcción habría finalizado. Lo siguiente ya es el Go live, que es el lanzamiento de la página web. Después de esto habría que crear otra fase que es la de mantenimiento en la que se crean nuevos sprints con las nuevas tareas a realizar. En este caso seguiríamos utilizando esta plataforma debido a su éxito.

Conclusión:

La idea principal de este apartado es ver como trabajamos con la herramienta de Monday.com gestionando un proyecto creado por nosotros. A nivel profesional y contando con una versión totalmente gratuita podemos decir que se trata de una plataforma que

se adapta a las necesidades de la empresa/proyecto que se esté gestionando ya que ofrece gran variedad de plantillas, ordenadas por categoría, y que te permite la gestión de cualquier proyecto, desde un CRM hasta un evento universitario. Los elementos colaborativos que nos ofrece la herramienta fomenta la comunicación entre los miembros del equipo ya que se pueden añadir comentarios, menciones, adjuntar archivos, números de teléfonos y correos. Tenemos que recalcar la capacidad que tiene Monday para recopilar toda la información que le introducimos y poder plasmarla en diferentes tipos de diagramas, así como exportar a excel esa información que introducimos en cualquier momento. De lo que hemos carecido a lo largo del desarrollo del proyecto es de las integraciones con otras aplicaciones/herramientas, y de la creación de automatizaciones. Esto se consigue actualizando el plan de Monday. La única automatización con la que hemos contado es que si una tarea cambiaba de estado de proceso, se creaba una actualización en la actividad del tablero. En el caso de contar con un plan más avanzado estas serían las automatizaciones e integraciones que ofrece Monday:

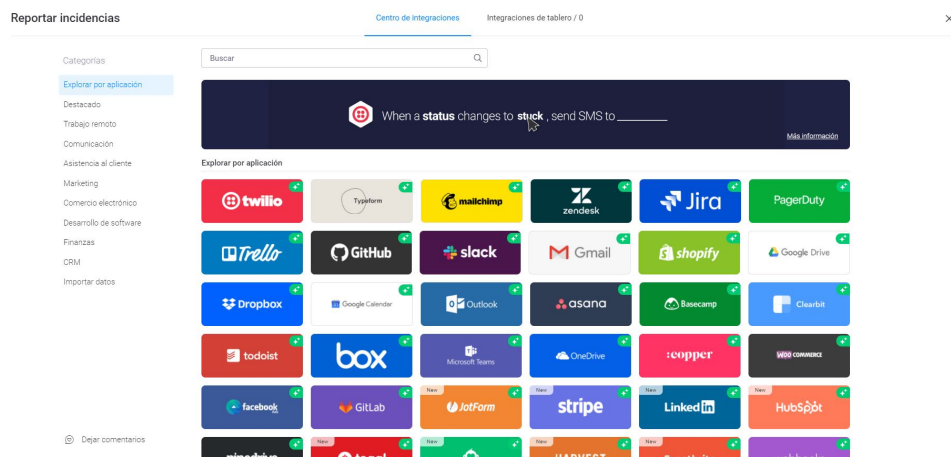


Figura 4.19: Integraciones con otras herramientas (A partir del plan estándar)

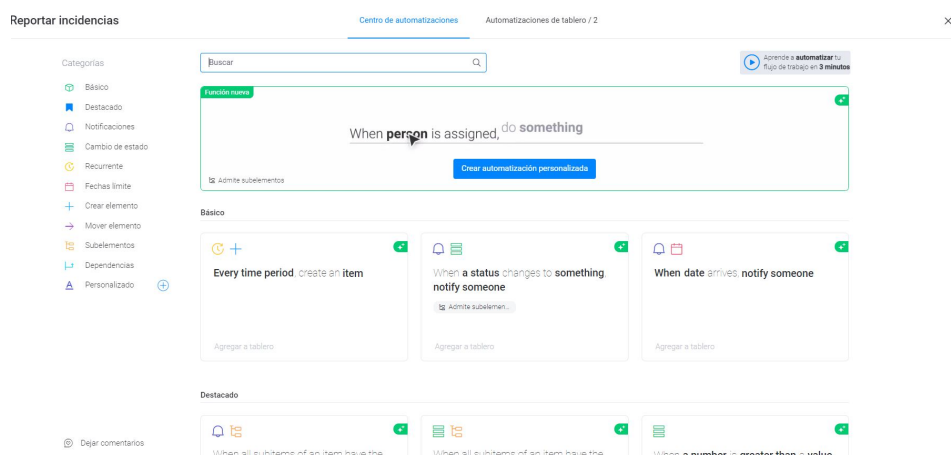


Figura 4.20: Automatizaciones con otras herramientas (A partir del plan estándar)

Por último me gustaría añadir como comentario personal que gracias a la interfaz tan ordenada, clara e intuitiva que tiene Monday, consigue que la organización sea mucho más fácil y podamos ahorrar energías para otras partes del proyecto que probablemente lo necesiten más. Pero, sin una buena base de organización no se puede construir nada en condiciones y Monday nos lo pone más fácil.

4.4 Asana

Asana es la segunda herramienta con la que vamos a gestionar el proyecto que hemos definido en el inicio del capítulo 4. Vamos a intentar realizar los mismos pasos que en Monday para poder comparar hasta que punto ofrecen las mismas funcionalidades y de qué manera. Es posible que Asana ofrezca otras funcionalidades que Monday no, o viceversa. Eso es lo que finalmente queremos comparar y ver cuál es más efectiva para nosotros.

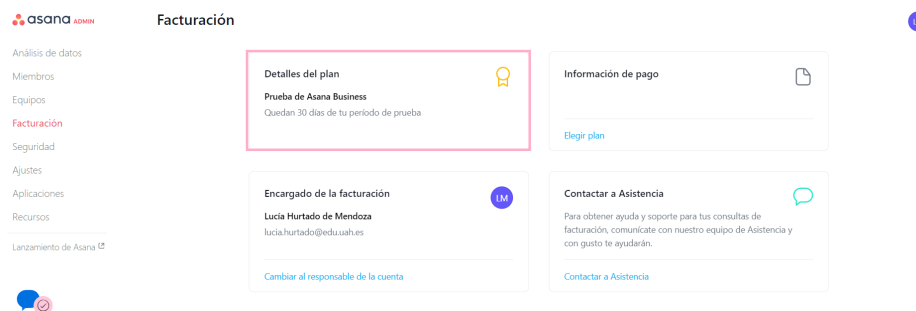


Figura 4.21: Plan Business de prueba gratuito

Lo primero de todo es aclarar que el plan que vamos a utilizar es la prueba gratuita del plan Business, que ofrece algunas funcionalidades más que el plan básico. Sin embargo, en Monday no teníamos la opción de probar gratis durante un periodo de tiempo el plan que quisiésemos, sino que nos ofrecía el plan para estudiantes con algunas funcionalidades más.

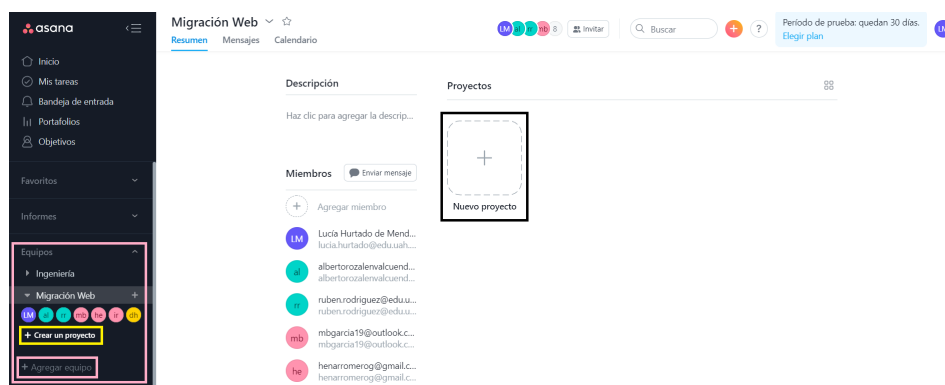


Figura 4.22: Creación del proyecto en Asana

Para crear un proyecto en Asana lo primero que hacemos es crear un equipo para que se quede fijo en el menú lateral izquierdo. Desde ahí invitamos a que se unan los miembros del equipo. Nuestro equipo lo hemos llamado 'Migración Web'. Después de crear el equipo procedemos a crear el proyecto, en el que Asana nos da la opción de crearlo a partir de unas plantillas ya preestablecidas.



Figura 4.23: Plantillas para la creación de un proyecto

Hemos elegido la plantilla de 'Planificación de sprints' que nos crea ya el tablero con los campos que Asana considera necesarios. En nuestro caso hemos eliminado la columna de costo, ya que nos parece un detalle prescindible. Hemos intentado añadir la columna de dependencias de tareas para que quede similar a la planificación en Monday pero no contamos con esa opción. Finalmente, la planificación de la fase de construcción en Asana quedaría en visualización 'Lista' tal y como en las figuras que se adjuntan a continuación. El proyecto se guarda dentro del equipo 'Migración Web' en el menú lateral izquierdo.

| Nombre de la tarea | Responsable | Fecha de ent... | Prioridad | Estado del pr... |
|--|-----------------|-----------------|-----------|------------------|
| Documento funcional | Lucía Hurtar... | 15 - 17 mar | Alta | Por hacer |
| U51 Desarrollo Backend de la cabecera | rubenrodri... | 18 - 19 mar | Alta | Hecho |
| U52 Desarrollo Frontend de la cabecera | mbgarcia19... | 22 - 23 mar | Alta | Hecho |
| U53 Desarrollo Backend del pie de página | henarome... | 18 - 19 mar | Alta | Hecho |
| U54 Desarrollo Frontend del pie de página | mbgarcia19... | 23 - 24 mar | Alta | Por hacer |
| U55 Desarrollo Backend del banner destacado | rubenrodri... | 22 - 23 mar | Alta | Por hacer |
| U56 Desarrollo Frontend del banner destacado | mbgarcia19... | 24 - 25 mar | Alta | Por hacer |
| Revisión | albertorcal... | 26 mar | Alta | Por hacer |

Figura 4.24: Sprint 1

| Nombre de la tarea | Responsable | Fecha de ent... | Prioridad | Estado del pr... |
|---|----------------|-----------------|-----------|------------------|
| U57 Desarrollo Backend de la tabla dinámica texto | henarome... | 29 - 31 mar | Alta | Hecho |
| U58 Desarrollo Frontend de la tabla dinámica texto | mbgarcia19... | 1 - 2 abr | Alta | Hecho |
| U59 Desarrollo Backend del módulo características | rubenrodri... | 29 - 30 mar | Alta | Hecho |
| U510 Desarrollo Frontend del módulo características | mbgarcia19... | 5 - 6 abr | Alta | Hecho |
| U511 Desarrollo Backend del módulo características | henarome... | 1 - 5 abr | Alta | Hecho |
| U512 Desarrollo Frontend del módulo características | rubenrodri... | 7 - 8 abr | Alta | Hecho |
| Revisión | albertorcal... | 9 abr | Alta | Hecho |

Figura 4.25: Sprint 2

| Nombre de la tarea | Responsable | Fecha de ent... | Prioridad | Estado del pr... |
|--|----------------|-----------------|-----------|------------------|
| U513 Subir contenido del cliente | henarome... | 12 abr | Alta | Hecho |
| U514 Posicionamiento SEO | dhurtadode... | 12 - 15 abr | Alta | Hecho |
| U515 Analítica | henarome... | 12 - 16 abr | Alta | Hecho |
| U516 Configuración de los componentes contribuyendo el contenido real del cliente que se ha subido | henarome... | 13 - 15 abr | Alta | Hecho |
| Revisión completa de la página web | albertorcal... | 19 - 21 abr | Alta | Hecho |

Figura 4.26: Sprint 3

Si queremos añadir más columnas tendremos que pulsar en el '+' que se encuentra a la derecha de la tabla y nos saldrán las opciones que tenemos. La opción de 'Desde la biblioteca' es una serie de columnas que ya están predefinidas por Asana, por si no se nos ocurre ninguna idea.

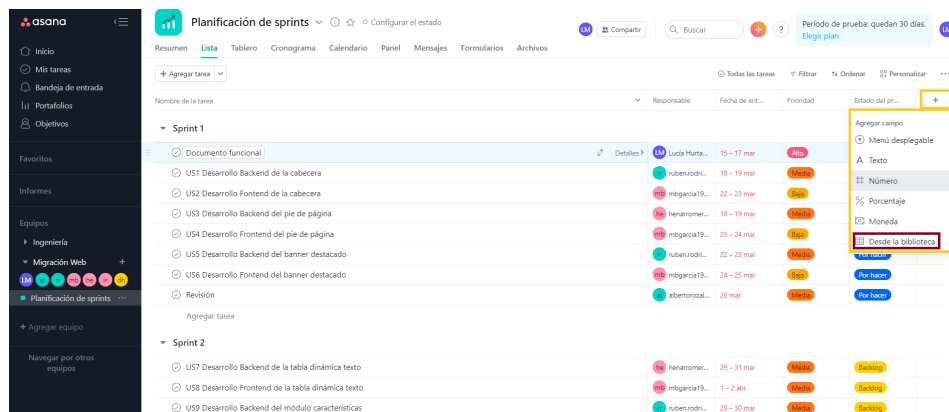


Figura 4.27: Agregar columnas a la tabla

En Asana podemos ver los detalles de cada tarea pinchando en cada una de ellas, entonces se nos abre un menú lateral a la derecha donde nos muestra toda la información que habíamos introducido. Además, desde aquí podemos crear subtareas que pueden tener una fecha de entrega, prioridad o estado diferente. También podemos adjuntar los documentos que queramos en cualquier tarea. En el caso de tener alguna duda o querer comentar algo se puede hacer directamente en el detalle de la actividad y mencionando a quien corresponda.

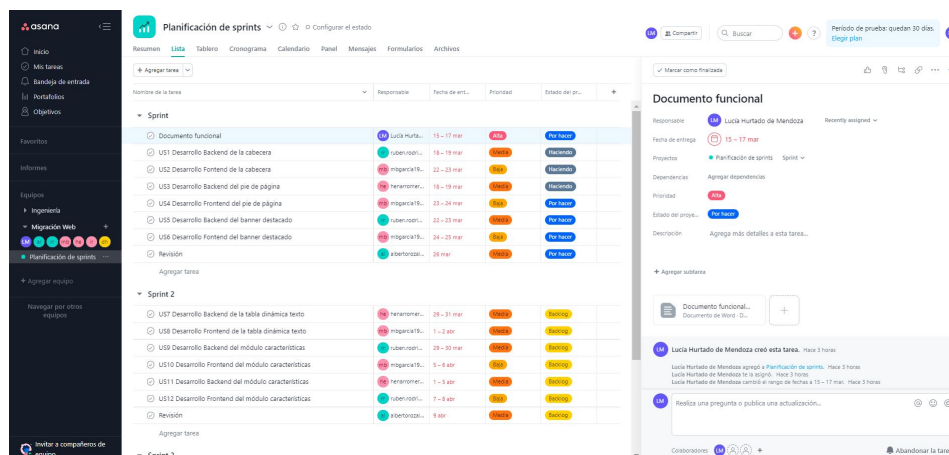


Figura 4.28: Detalle de las tareas de los sprints

Una muy buena funcionalidad que considero que ofrece Asana es que si la actividad la marcamos como 'tarea de seguimiento' se nos añade al menú de 'mis tareas'. Así podemos ver todas las que tenemos asignadas, siendo mucho más fácil diferenciar nuestras tareas de la de los demás. Además, pulsando en la opción de 'Mis tareas' podemos ver el proyecto al que pertenece las tareas que tenemos asignadas y las que marcamos para hacer un seguimiento.

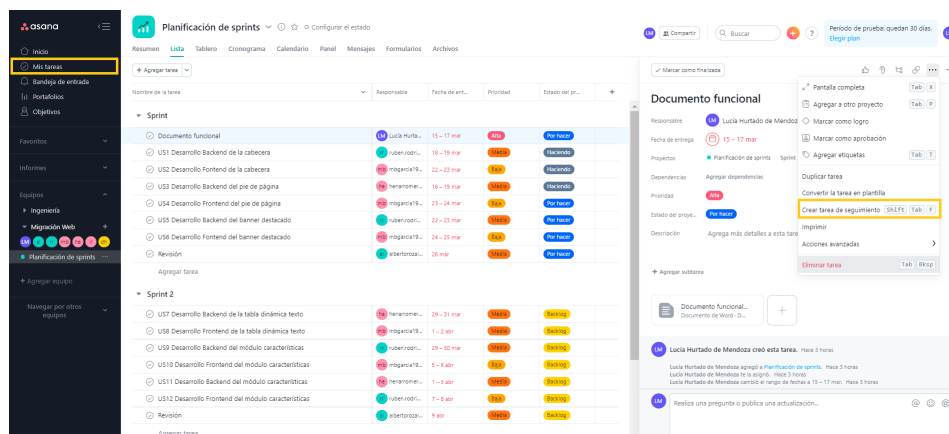


Figura 4.29: Marcar tarea de seguimiento

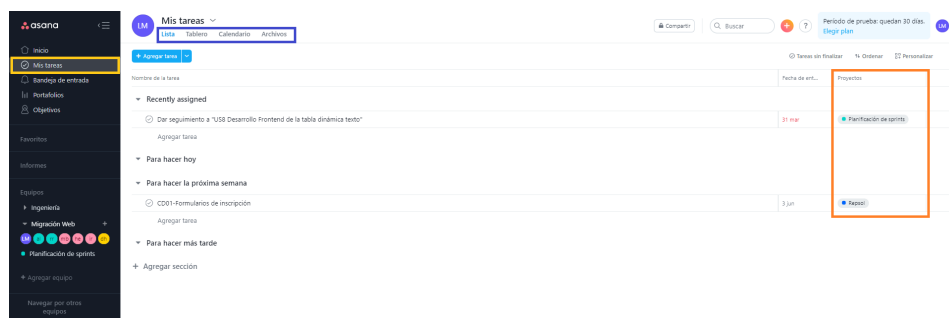


Figura 4.30: Mis tareas

En cuanto a las vistas que ofrece Asana también son bastantes. La que hemos estado viendo durante el proyecto era la visualización tabla. Pero, también tenemos otras visualizaciones como la del tablero.

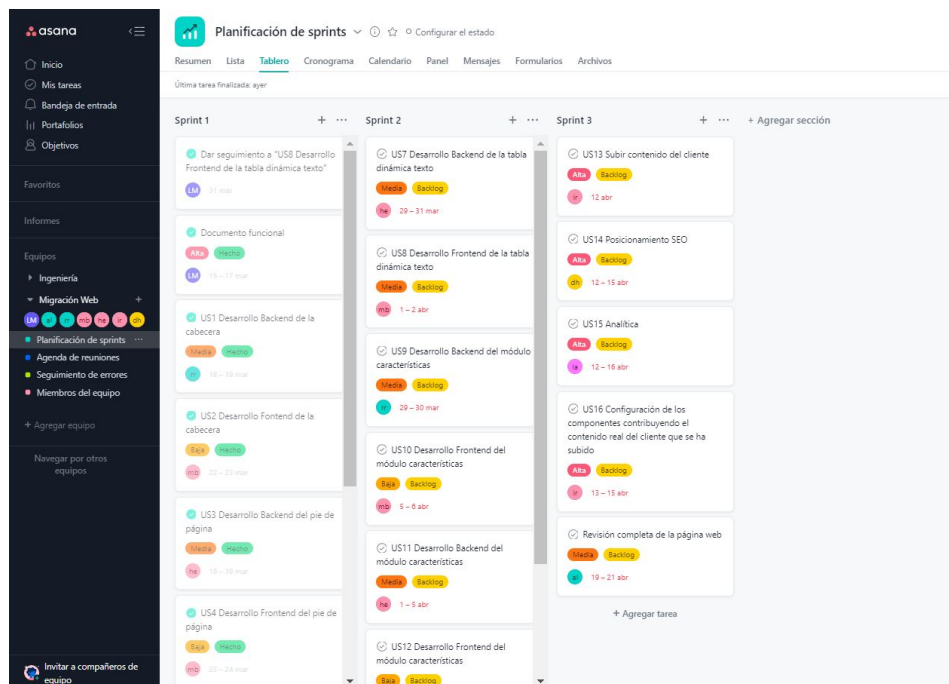


Figura 4.31: Vista tablero

Como podemos ver, no se trata de un tablero Kanban en el que las tareas se agrupen por el estado en el que se encuentran, sino por Sprint que es como se ha organizado en esta plantilla. Por lo que hay dos opciones, si queremos organizarlo por estado de la tarea habría que cambiar el nombre de los grupos/columnas en vez de 'Sprint x' a 'Por hacer, haciendo y hecho', todas las tareas de los sprints estarían juntas sin poder diferenciarse de un sprint u otro o, la otra opción es crear un tablero por cada Sprint. Esto lo consideramos una desventaja ya que no es tan visual como en Monday y puede llevar a confusiones.

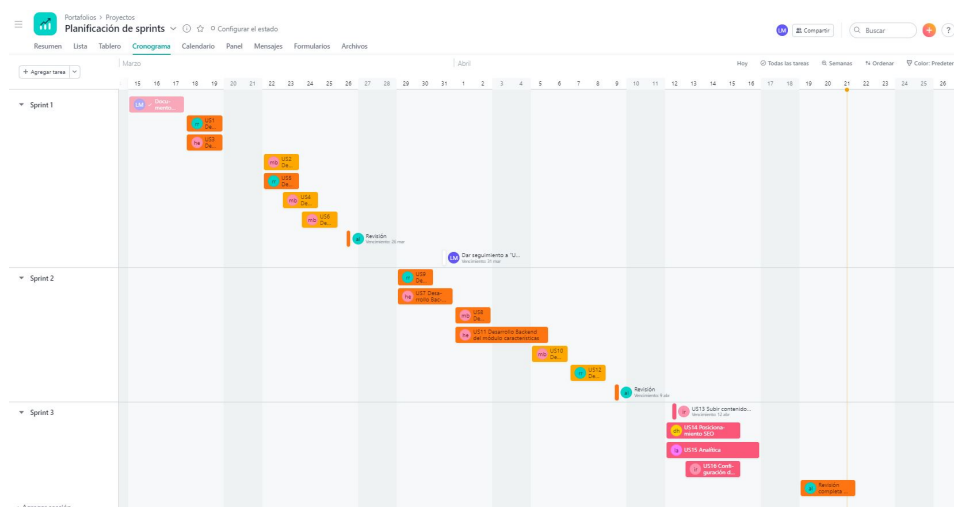


Figura 4.32: Vista cronograma

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A15 | A16 | A17 | A18 | A19 | A20 | A21 | A22 | A23 |
|------------------|------------|--------------|---------------|--|----------------|--------------------------|---------------------------------|------------|------------|------|-------|----------|-------------|----------|---------------------|-----|-----|-----|-----|-----|-----|-----|
| Task ID | Created At | Completed At | Last Modified | Name | Section/Column | Assignee | Assignee Email | Start Date | Due Date | Tags | Notes | Projects | Parent Task | Priority | Estado del proyecto | | | | | | | |
| 1200260294431598 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend de la tabla dinámica texto | Sprint 1 | Lucia Hurtado de Mendoza | lucia.hurtado@edu.uah.es | 2021-03-31 | 2021-03-31 | | | | | | | | | | | | | |
| 1200260294431444 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Documento funcional | Sprint 1 | Lucia Hurtado de Mendoza | lucia.hurtado@edu.uah.es | 2021-03-15 | 2021-03-17 | | | | | | | | | | | | | |
| 1200260294431458 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Backend de la cabecera | Sprint 1 | Ruben Rodriguez | ruben.rodriguez@edu.uah.es | 2021-03-18 | 2021-03-19 | | | | | | | | | | | | | |
| 1200260294431460 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend de la cabecera | Sprint 1 | Miguel Garcia | mbgarcia19@outlook.com | 2021-03-22 | 2021-03-23 | | | | | | | | | | | | | |
| 1200260294431462 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Backend del pie de página | Sprint 1 | Henarromero | henarromero@gmail.com | 2021-03-18 | 2021-03-19 | | | | | | | | | | | | | |
| 1200260294431464 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del pie de página | Sprint 1 | Miguel Garcia | mbgarcia19@outlook.com | 2021-03-23 | 2021-03-24 | | | | | | | | | | | | | |
| 1200260294431466 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Backend del banner destacado | Sprint 1 | Ruben Rodriguez | ruben.rodriguez@edu.uah.es | 2021-03-22 | 2021-03-23 | | | | | | | | | | | | | |
| 1200260294431470 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Revisión | Sprint 1 | Alberto Zalevalcuende | alberto.zalevalcuende@gmail.com | 2021-03-26 | 2021-03-26 | | | | | | | | | | | | | |
| 1200260294431472 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Backend de la tabla dinámica texto | Sprint 2 | Henarromero | henarromero@gmail.com | 2021-03-29 | 2021-03-31 | | | | | | | | | | | | | |
| 1200260294431474 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend de la tabla dinámica texto | Sprint 2 | Miguel Garcia | mbgarcia19@outlook.com | 2021-04-01 | 2021-04-02 | | | | | | | | | | | | | |
| 1200260294431476 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Backend del módulo características | Sprint 2 | Ruben Rodriguez | ruben.rodriguez@edu.uah.es | 2021-03-29 | 2021-03-30 | | | | | | | | | | | | | |
| 1200260294431478 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del módulo características | Sprint 2 | Miguel Garcia | mbgarcia19@outlook.com | 2021-04-05 | 2021-04-06 | | | | | | | | | | | | | |
| 1200260294431480 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Backend del módulo características | Sprint 2 | Henarromero | henarromero@gmail.com | 2021-04-01 | 2021-04-05 | | | | | | | | | | | | | |
| 1200260294431482 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del módulo características | Sprint 2 | Ruben Rodriguez | ruben.rodriguez@edu.uah.es | 2021-04-07 | 2021-04-08 | | | | | | | | | | | | | |
| 1200260294431484 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Revisión | Sprint 2 | Alberto Zalevalcuende | alberto.zalevalcuende@gmail.com | 2021-04-09 | 2021-04-09 | | | | | | | | | | | | | |
| 1200260294431486 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del cliente | Sprint 3 | Jeneprieto | jeneprieto14@gmail.com | 2021-04-12 | 2021-04-12 | | | | | | | | | | | | | |
| 1200260294431488 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del cliente | Sprint 3 | Jeneprieto | jeneprieto14@gmail.com | 2021-04-12 | 2021-04-12 | | | | | | | | | | | | | |
| 1200260294431490 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del cliente | Sprint 3 | Jeneprieto | jeneprieto14@gmail.com | 2021-04-12 | 2021-04-12 | | | | | | | | | | | | | |
| 1200260294431492 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Desarrollo Frontend del cliente | Sprint 3 | Jeneprieto | jeneprieto14@gmail.com | 2021-04-12 | 2021-04-12 | | | | | | | | | | | | | |
| 1200260294431494 | 2021-04-28 | 2021-04-28 | 2021-04-28 | Revisión completa de la página web | Sprint 3 | Alberto Zalevalcuende | alberto.zalevalcuende@gmail.com | 2021-04-19 | 2021-04-21 | | | | | | | | | | | | | |

Figura 4.33: Planificación de Sprints exportado a excel

La vista cronograma en Asana nos recuerda al diagrama de Gantt, pero no podemos decir que sea el diagrama de Gantt ya que existe ningún tipo de vinculación entre las tareas debido a que no podemos establecer sus dependencias en esta herramienta. Permite exportar el proyecto tanto a CSV como a JSON, aunque el resultado que obtenemos no sea muy visual.

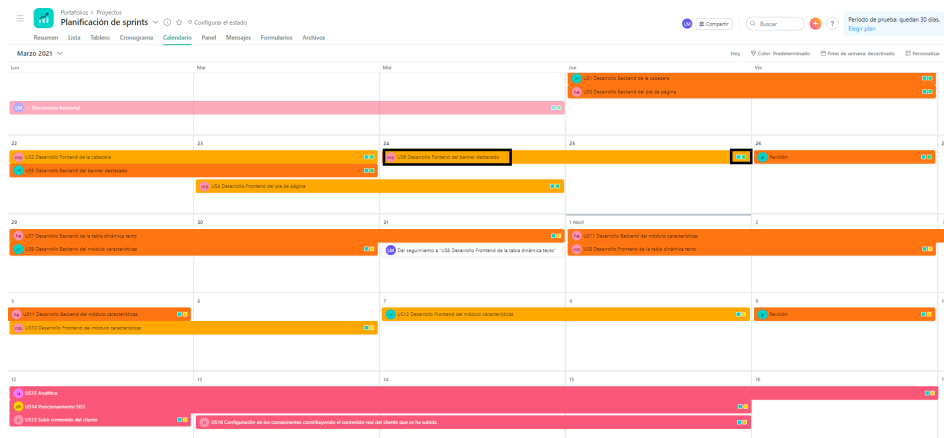


Figura 4.34: Calendario

Por último, la vista calendario muestra la duración de las tareas con la franja del color de la prioridad que tienen cada una, el nombre de la tarea, quien la tiene asignada y en la esquina derecha de cada franja hay dos cuadrados que representan el proyecto a la que pertenece esa tarea y el estado de la misma. Bajo mi punto de vista, esta visualización no es nada práctica.

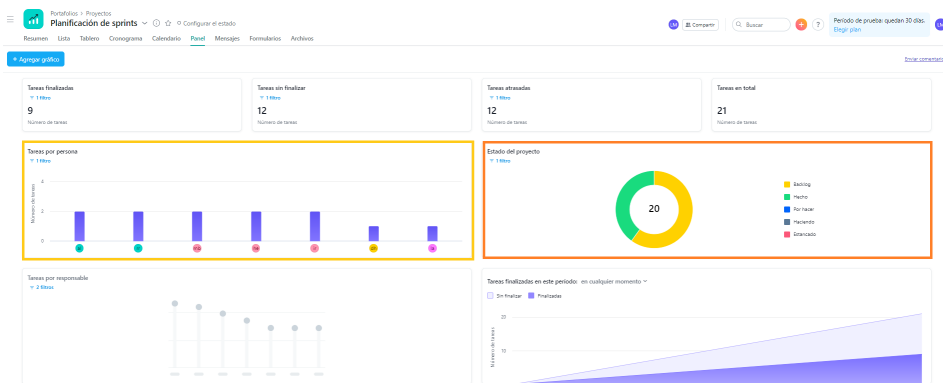


Figura 4.35: Paneles

Cuando ya hemos completado alguna de las tareas gracias al esfuerzo de los miembros del equipo, los marcamos con la etiqueta de 'Hecho' y marcamos un check que hay en cada tarea. Hasta ahora hemos completado el Sprint 1. A continuación, vamos a utilizar los paneles que ofrece Asana para ver el estado de nuestro proyecto en un gráfico y poder interpretarlo más fácilmente. El primer gráfico muestra las tareas por miembro del equipo y el segundo muestra el estado del proyecto en general. Los otros dos paneles nos aparecerán si seguimos trabajando con la herramienta.

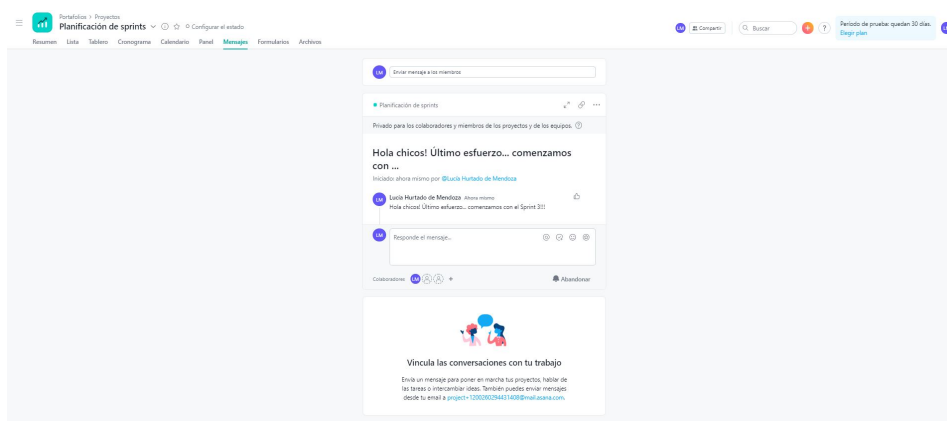


Figura 4.36: Chat de Asana

Además, Asana cuenta con un chat colectivo en el que todos los miembros del equipo pueden participar. Es muy importante porque cualquier comentario que tengamos acerca del proyecto podemos publicarlo en este chat y ser respondidos por cualquier compañero que participe en este proyecto. Podemos utilizarlo también para enviar mensajes que motiven al equipo, o dar las gracias y la enhorabuena por su colaboración y esfuerzo a todos, ya que mensajes así halagan a cualquiera.

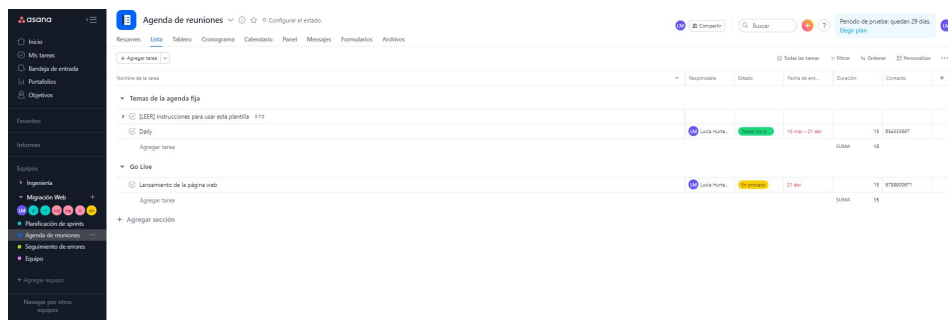


Figura 4.37: Agenda de reuniones

Las reuniones que se realizan durante el proyecto las hemos organizado en un tablero con la plantilla que nos ofrece Asana llamada 'Agenda de reuniones'. El tablero no queda tan visual como en el caso de Monday, pero sirve para organizarnos.

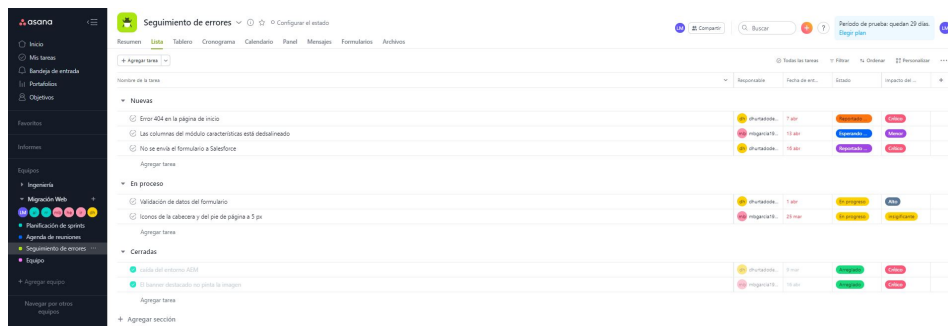


Figura 4.38: Seguimiento de errores

En cuanto a las incidencias que han ido surgiendo durante el proyecto se han reportado en la plantilla 'Seguimiento de errores', en la que se plasma perfectamente los detalles de cada incidencia.

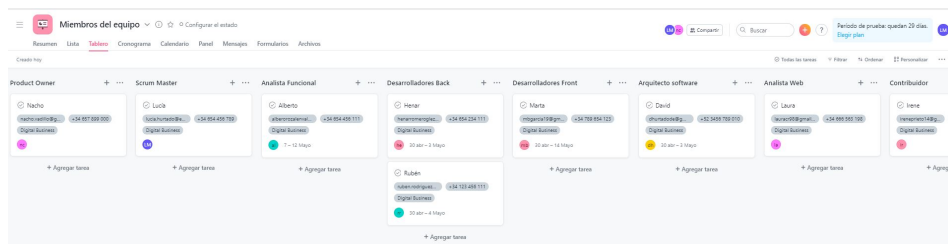


Figura 4.39: Información de los miembros del equipo

Por último, queríamos guardar la información de cada uno de los miembros del equipo para poder contactar con la persona que necesitemos en caso de alguna emergencia o duda importante sobre el proyecto. Hemos investigado las plantillas que nos ofrecía Asana y hemos probado alguna de ellas antes de crearnos el tablero nosotros desde cero. Finalmente, hemos optado por la plantilla 'Incorporación de un empleado nuevo' porque podemos añadir y eliminar columnas para adaptarlo a nuestro gusto. Lo hemos querido mostrar en la versión tablero ya que la visualización es mucho más limpia que en lista.

Una vez ya organizado el proyecto, cada miembro del equipo tiene que estar atento a las notificaciones de la plataforma y cumplir con las tareas que se le han asignado, así como a las fechas de entrega. Cuando se cumplen, se marca la etiqueta de 'Hecho' hasta finalizar todos los sprints. El procedimiento que se sigue cuando se finaliza la fase de construcción es el mismo que hemos detallado en Monday. Se hace el lanzamiento de la página y se prepara la fase de mantenimiento.

Conclusión:

Nuestra experiencia con Asana ha cumplido las expectativas. Queríamos comentar en general que esta plataforma también ofrece gran cantidad de funcionalidades. Una de las funcionalidades que nos parece más útiles son las plantillas, ya que nos guían sobre qué campos utilizar para introducir la información que necesitemos para hacer un buen seguimiento del proyecto. Al igual que Monday, tiene mucha variedad en este aspecto y se organizan por categorías. Esta plataforma ofrece integraciones con otras aplicaciones/herramientas, así como automatizaciones, que en el caso de Asana se llaman reglas. Para añadir estas reglas lo tenemos que hacer desde el proyecto que queramos y pinchar en la opción personalizar, desde donde podemos añadir tanto las integraciones como las reglas.

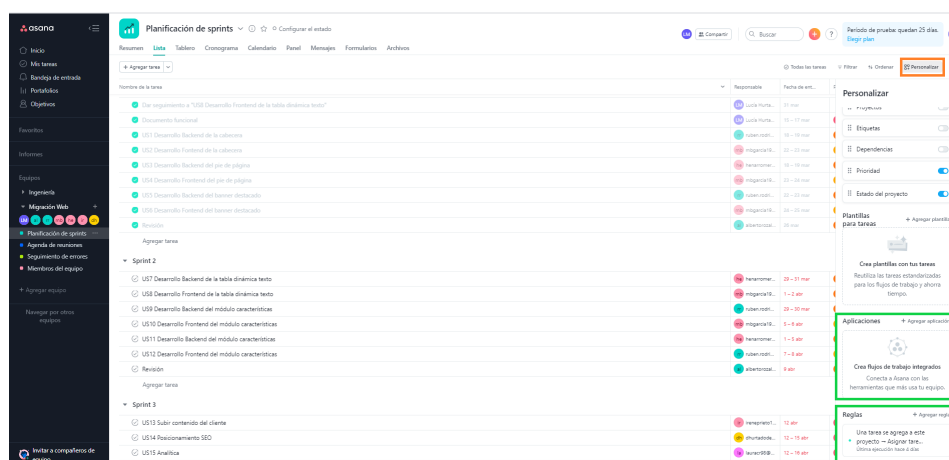


Figura 4.40: Personalizar proyecto

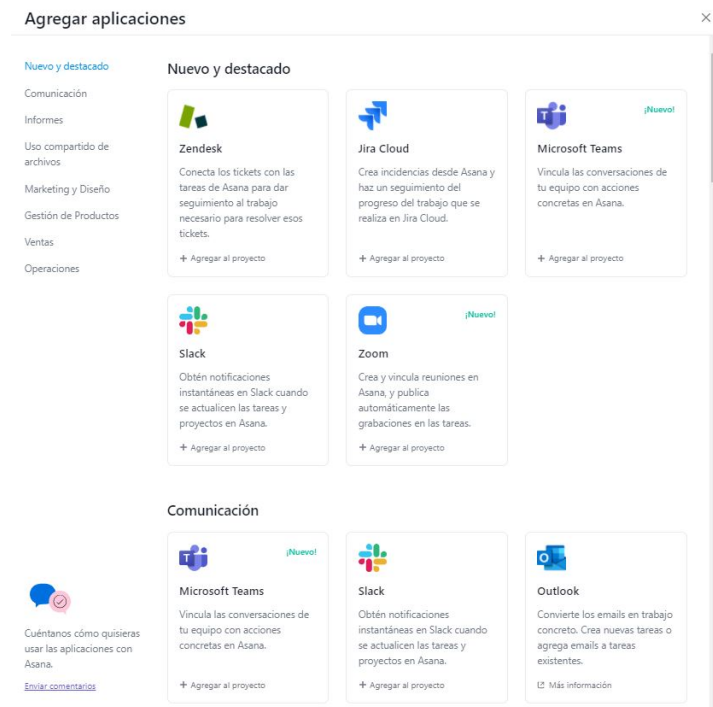


Figura 4.41: Integraciones con herramientas

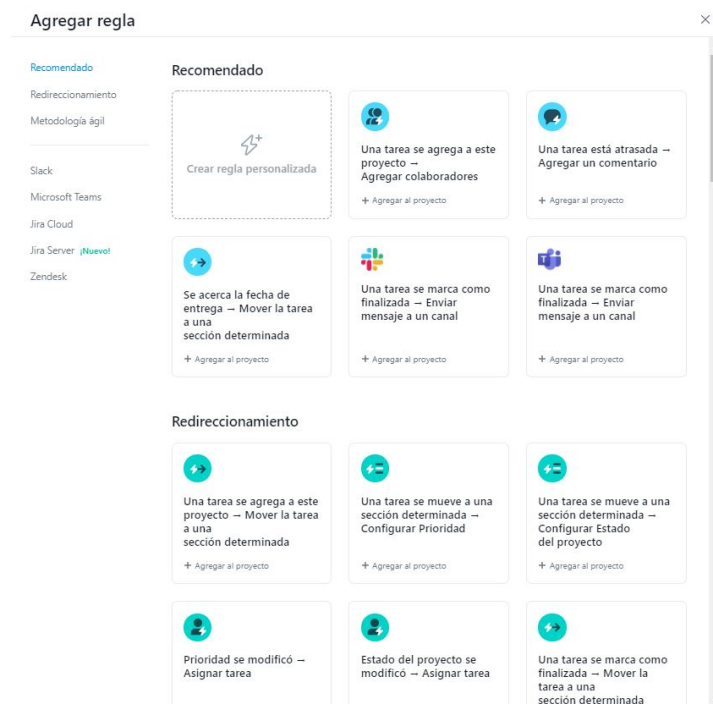


Figura 4.42: Reglas de Asana

Por último, comentar que se trata de una interfaz intuitiva en la que tenemos a disposición las funcionalidades necesarias para gestionar y hacer un seguimiento del

proyecto en cuestión. Por otro lado, remarcar que la solución colaborativa de Asana me parece muy buena ya que está a disposición de todos los miembros del equipo y se pueden añadir tanto documentos como menciones.

4.5 Planteamiento final

En base al análisis realizado en los capítulos anteriores, la decisión final recogerá tanto los criterios funcionales como los criterios personales. Haciendo una recomendación de uso de las herramientas de gestión de proyectos donde nos basamos tanto en criterios objetivos como subjetivos.

Criterios funcionales

La funcionalidad de las herramientas las hemos tenido claras desde el principio, ya que las hemos analizado en el capítulo 3. Es verdad que en Monday hemos utilizado el plan para estudiante, pero podíamos probar algunas funcionalidades que nos ofrecían durante 14 días, entre ellas el diagrama de Gantt. Sin embargo, en Asana hemos utilizado el plan business, que durante 14 días nos ofrecía algunas funcionalidades más que el plan básico, por ejemplo el diagrama de Gantt. Es decir, ambas nos ofrecen las mismas funcionalidades de prueba, por lo que en cuanto a este aspecto no hay ventaja de una herramienta sobre otra, sino en cuanto a la usabilidad.

Teniendo en cuenta el nivel funcional que proporcionan, la facilidad de uso y el interfaz de usuario, en definitiva cómo el usuario se siente al utilizar la herramienta, ambas son equivalentes y están a la par.

Criterios personales

Como experiencia de usuario la herramienta que más me ha gustado es la de Monday, ya que me parece un poco más organizada que Asana y me ha costado menos familiarizarme con ella. Además, como plasma la información en las distintas visualizaciones y en los diagramas, es estéticamente más fácil de entender para los miembros del equipo. En términos generales Monday es más ordenada, detallista y fácil de aprender, ya que muchos elementos son autoexplicativos.

Desde el punto de vista personal, teniendo en cuenta que funcionalmente aportan lo mismo, mi decisión sería utilizar Monday.

Capítulo 5

Conclusiones

Como conclusión del TFG "Herramientas para la planificación y el seguimiento de proyectos ágiles" destaco que se ha conseguido los objetivos propuestos al inicio del trabajo. En este proyecto de fin de carrera he querido mostrar la evolución de las metodologías ágiles, diferenciándolas de las tradicionales, así como su constante crecimiento.

Durante todo el documento se ha querido destacar la importancia que tiene planificar bien un proyecto, evitando crear algunas circunstancias que pueden conducir al retraso de este. No sólo hemos querido mostraros la parte teórica de estas metodologías ágiles, sino que hoy en día contamos con gran cantidad de herramientas que nos ayudan a planificar estos proyectos, pero primero debemos partir desde la base. Como ya hemos contado, existen muchas herramientas en el mercado actual, de las cuáles hemos querido hacer una selección para analizarlas y saber cuál se podría adaptar más a nuestras condiciones o necesidades. Hemos considerado que lo mejor para plasmar lo aprendido es aplicarlo. Para ello hemos simulado un proyecto y lo hemos llevado a cabo con Monday.com y Asana.

La elaboración de este proyecto me ha llevado a conocer algunas dificultades a la hora de gestionar un proyecto y saber aplicar las buenas prácticas de las metodologías ágiles tanto al proyecto como al equipo. Es una pena que sólo haya sido un simulacro porque estoy segura de que esto en la vida real sería aún más complicado sólo por el hecho de que hay factores externos que nos pueden retrasar en el proyecto y tengamos que tomar decisiones sobre la marcha. Otra de las ventajas que tiene utilizar estas metodologías es que una vez planificado un proyecto podemos reutilizar la plantilla que hemos creado haciendo algunas modificaciones. Siempre y cuando el proyecto sea similar, el tiempo en organizar el proyecto será menor porque reutilizaremos esa plantilla y no tendremos que reinventar todo desde cero. Hablo por ejemplo de las plantillas que nos han ofrecido en todo momento Monday.com y Asana según la categoría a la que estuviese enfocado el proyecto. Hemos tenido que tomar una decisión en cuanto a la elección de la herramienta que gestionará nuestros futuros proyectos, y esta ha sido Monday.com, ya que es completa en todos los aspectos que hemos requerido a lo largo de esta memoria.

Finalmente, desear al lector de este documento que encuentre las buenas prácticas de las metodologías ágiles a la hora de planificar un proyecto, así como la herramienta de

gestión que más confianza le transmita tanto a él como a su equipo y que cubra las necesidades de su proyecto.

Capítulo 6

Trabajos Futuros

Tras haber cumplido con los objetivos de este TFG, se proponen los siguientes trabajos y mejoras relacionados con el desarrollo ágil de software:

- **Desarrollar el mismo proyecto en el resto de herramientas analizadas:** no todas los usuarios nos basamos en el mismo criterio para definir una herramienta como la mejor elección para gestionar nuestros proyectos, por lo que invito a otro usuario que desarrolle el mismo proyecto en el resto de plataformas y realice la comparación con este trabajo.
- **Continuar con la fase de mantenimiento:** seguir desarrollando el proyecto con la fase de mantenimiento, así podemos probar otras plantillas diferentes que no hayan sido utilizadas y también ver como se comporta la plataforma que utilizamos cuando tiene más información de un proyecto, así como la creación de informes/diagramas sobre el proyecto en conjunto.
- **Investigar otras metodologías ágiles:** durante este documento hemos investigado algunas metodologías ágiles. Básicamente las consideradas más utilizadas o que más auge tienen hoy en día. Como propuesta se puede investigar nuevas metodologías y enfocar un proyecto con una de las que hayamos encontrado.

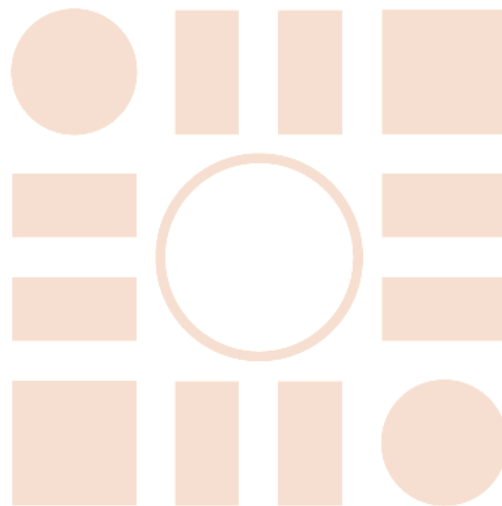
Bibliografía

- [1] Encarna Abellán. Metodología scrum: qué es y cómo funciona. *wam*, March 2020.
- [2] Digital Talen Agency. Modelo waterfall o en cascada. Technical report, 2018.
- [3] Agile Alliance. Programación extrema (xp). Disponible en: <https://bit.ly/3rMLWvj>.
- [4] Antevenio. ¿qué es la metodología scrum? guía práctica con ejemplos. Disponible en: <https://bit.ly/3bHTutU>, February 2020.
- [5] Asana. Conoce las funciones principales de asana. Disponible en: <https://bit.ly/3cVJw7G>.
- [6] Atlassian. Empezar con jira software. Disponible en: <https://bit.ly/2OCuMlZ>, 2021.
- [7] M. Awad. Comparison between agile and traditional software development methodologies. 2005.
- [8] Kent Beck. Extreme programming explained. In *Embrace change*, September 2004.
- [9] Capterra. Monday.com. Disponible en : <https://bit.ly/3u5ifX5>.
- [10] Capterra. Wrike. Disponible en: <https://bit.ly/3cVabBr>.
- [11] Capterra. Software de gestión de proyectos. Disponible en: <https://bit.ly/2QaqHWr>, December 2020.
- [12] Capterra. Software de gestión del ciclo de vida de las aplicaciones. Disponible en: <https://bit.ly/3r1Qi0C>, July 2020.
- [13] Ward Cunningham. Manifesto for agile software development, 2001.
- [14] Dr. Dana. Tablero kanban: no es la lista promedio de tareas pendientes. *Fedability*, March 2018.
- [15] Matteo Duò. Trello vs asana – ¿dónde administrar sus proyectos en 2021? *KINSTA BLOG*, December 2020.

- [16] EcuRed. Wrike. Disponible en: <https://www.ecured.cu/Wrike>.
- [17] Software Empresarial. Análisis del software jira para gestionar proyectos. Disponible en: <https://bit.ly/2OWK4BG>, 2021.
- [18] Shabib Aftab Usman Waheed Faiza Anwer, Syed Shah Muhammad. Comparative analysis of two popular agile processmodels: Extreme programming and scrum. *International Journal of Computer Science and Telecommunications*, 8, March 2017.
- [19] Johana Zumba Gamboa. Evolución de las metodologías y modelos utilizados en el desarrollo de software. *Dialnet*, 2018.
- [20] Javier Garzas. ¿cuál es el estado de la agilidad a nivel mundial? 14th state of agile, 2020.
- [21] Demián Gutierrez. Metodologías y procesos de análisis de software., July 2011.
- [22] Darío Herrera. La historia de scrum. *Dario Herrera*, December 2017.
- [23] Kysbel Hidalgo. Metodología kanban: El método japonés que agilizará tu negocio. *Mente diamante*, February 2020.
- [24] Jim Highsmith. *Agile Software Development Ecosystems*, chapter Preface, page 389. Pearson Education Corporate Sales Division, 2002.
- [25] Ron Jeffries. What is extreme programming? *RonJeffries.com*, March 2011.
- [26] KanbanTool. La historia de kanban. Disponible en: <https://bit.ly/3tjGuk5>, 2009.
- [27] Jeff Sutherland Ken Schwaber. La guía de scrum. In *La Guía Definitiva de Scrum: Las Reglas del Juego*, 2017.
- [28] Alex Konrad. El unicornio de israel: un administrador de proyectos que vale 1,900 mdd. *Forbes*, July 2019.
- [29] Allen Lee. 10 cosas que no sabías sobre el ceo de monday, roy mann. *Money Inc*, 2020.
- [30] Knut Linke. Traditional and agile management approaches. 09 2019.
- [31] Maryam Alian Mahdi Javanmard. Comparison between agile and traditional software development methodologies. *Science Journal*, 2015.
- [32] Scrum Manager. Origen de scrum. Disponible en: <https://bit.ly/3ln72OD>, April 2014.
- [33] Marvin López Mendoza. Extreme programming: Qué es y cómo aplicarlo. *Open-Webinars*, September 2020.

- [34] Monday.com. Descripción general del producto. Disponible en : <https://monday.com/>.
- [35] Yanina Muradas. Qué es trello. *OpenWebinars*, July 2020.
- [36] Finances Online. Project management software. Disponible en: <https://bit.ly/3gyJo1d>, April 2021.
- [37] Visual Paradigm. What are scrum ceremonies? Disponible en: <https://bit.ly/3ezP2iN>, 2020.
- [38] Arie van Bennekum Ryan Lockard James Gifford Jürgen Münch Michael Stuppersch Kurt Schneider Philipp Hohl, Jil Klünder. Origins and directions of the agile manifesto. *Journal of Software Engineering, Research and Devolpment*, 2018.
- [39] Julio Roche. Historia del movimiento agile, i parte. *Deloitte*, 2021.
- [40] Julio Roche. Historia del movimiento agile, ii parte. *Deloitte*, 2021.
- [41] Trello. Trello 101. Disponible en: <https://bit.ly/2P8Gnca>.
- [42] Eliécer Herrera Uribe. Del manifiesto ágil sus valores y principios. *Dialnet*, 2007.
- [43] Don Wells. Programación extrema: una suave introducción. *Agile Process*, October 2013.
- [44] Wrike. Transforme el trabajo con las funciones líderes del sector de wrake. Disponible en: <https://bit.ly/2Qqc0Pb>.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá